		<u>1</u> _
1	KEKER & VAN NEST, LLP JOHN W. KEKER - #49092	
2	MICHAEL H. PAGE - #154913 710 Sansome Street	•
3	San Francisco, CA 94111-1704	
4	Telephone: (415) 391-5400 Facsimile: (415) 397-7188	
5	INTERTRUST TECHNOLOGIES CORPORA	TION
6	DOUGLAS K. DERWIN - #111407 JEFFERY J. McDOW - #184727	
7	4800 Patrick Henry Drive Santa Clara, CA 95054	
8	Telephone: (408) 855-0100 Facsimile: (408) 855-0144	
9	PENNIE & EDMONDS LLP	
10	MICHAEL J. LYONS - #202284 300 Hillview Avenue	
11	Palo Alto, CA 94304 Telephone: (650) 493-4935	
12	Facsimile: (650) 493-5556	
13	Attorneys for Plaintiff and Counter-Defendant INTERTRUST TECHNOLOGIES CORPORA	TION
14		
15		S DISTRICT COURT
16	NORTHERN DISTR	UCT OF CALIFORNIA
17		l a vi covicación actin
18	INTERTRUST TECHNOLOGIES CORPORATION, a Delaware corporation,	Case No. C 01-1640 SBA (MEJ)
19	Plaintiff,	Consolidated with C 02-0647 SBA
20	v.	INTERTRUST'S DISCLOSURES OF ASSERTED CLAIMS AND
21	MICROSOFT CORPORATION, a	PRELIMINARY INFRINGEMENT CONTENTIONS PURSUANT TO
22	Washington corporation,	PATENT LOCAL RULES 3-1 and 3-2
23	Defendant.	('683, '193, '861, '721, '891, '900, '912, '019,
24	AND COUNTER ACTION.	'876, '181, and '402 Patents)
25		
j		·
26	,	
27		
28		

Pursuant to the Court's August 8, 2003 Order, Plaintiff InterTrust Technologies

Corporation ("InterTrust") hereby submits its Disclosures of Asserted Claims and Preliminary

Infringement Contentions under Patent Local Rules 3-1 and 3-2 ("PLR 3-1 & 3-2 Disclosures")

to Defendant Microsoft Corporation ("Microsoft"). These PLR 3-1 & 3-2 Disclosures supercede
all previous PLR 3-1 and PLR 3-2 disclosures served by InterTrust in this case.

PATENT LOCAL RULE 3-1: DISCLOSURE OF ASSERTED CLAIMS AND PRELIMINARY INFRINGEMENT CONTENTIONS

#### (a) Asserted claims

InterTrust currently contends that the Microsoft products identified herein infringe the claims of U.S. Patents Nos. 6,185,683 B1 ("the "683 patent"); 6,253,193 B1 ("the '193 patent"); 5,920,861 ("the '861 patent"); 6,157,721 ("the '721 patent"); 5,982,891 ("the '891 patent"); 5,892,900 ("the '900 patent"); 5,917,912 ("the '912 patent"); 5,915,019 ("the '019 patent"); 5,949,876 ("the '876 patent"); 6,112,181 ("the '181 patent"); and 6,389,402 B1 ("the '402 patent"), as identified in the attached claim charts. As discovery progresses, InterTrust may determine that additional Microsoft products infringe the asserted patents and/or that Microsoft infringes additional patent claims. InterTrust reserves the right to supplement and/or amend its disclosures and infringement contentions.

#### (b) Accused products

InterTrust contends that various Microsoft products infringe the patent claims identified in the claim charts attached hereto. Accused products are listed in Exhibit A hereto. Accused products are listed in Exhibit A hereto, which is intended to encompass past, present, and future product versions that include the accused features and/or functionality.

#### (c) Claim charts

InterTrust submits the attached claim charts based solely on information available to it to date. Discovery is ongoing, and additional information is likely to be produced during discovery. InterTrust therefore reserves the right to supplement and/or amend its infringement assertions as discovery proceeds.

11-

12

13

1415

16

17

18

19 20

21

22

2324

. 25

26

27

28

InterTrust contends that Microsoft infringes at least the claims of the '683, '193, '861, '721, '891, '900, '912, '019, '876, '181, and '402 patents identified in the claim charts attached hereto as Exhibits B and C:<sup>1</sup>

### (d) Literal infringement and the doctrine of equivalents

InterTrust contends that Microsoft infringes the claims of the '683, '193, '861, '721, '891, '900, '912, '019, '876, '181, and '402 patents as specified in Exhibits B and C both literally and under the doctrine of equivalents.

### (e) Priority from earlier applications

InterTrust claims priority for the claims of the '891, '912, '683, '193, '019, '876, and '402 patents-in-suit dating to application No. 08/388,107, filed February 13, 1995. InterTrust claims priority for the claims of the '900 patent-in-suit dating to application No. 08/695,927, filed August 12, 1996. InterTrust does not claim priority for the claims of the '721, '861, and '181 patents-in-suit dating to any earlier application.

### (f) Reliance on InterTrust's own products

InterTrust does not currently intend to rely on the assertion that its own Commerce and Rights System products practice at least some of the claimed inventions of the '683, '193, '861, '721, '891, '900, '912, '019, '876, '181, and '402 patents-in-suit to support its infringement assertions against Microsoft.

#### PATENT LOCAL RULE 3-2: DOCUMENT PRODUCTION ACCOMPANYING DISCLOSURE

### (a) Documents re disclosure and/or offer of sale

InterTrust is not currently aware of such documents other than the documents that have previously been produced. See 1T00017664-19168, 1T00020866-21695, 1T00021700-23578,

Exhibit B contains claim charts based upon publicly available or non-confidential sources. Exhibit C contains additional claim charts referencing material designated as "Attorneys' Eyes Only" by Microsoft, and is served under separate caption. No other information contained in these disclosures is designated confidential by either party, and InterTrust does not object to dissemination of this document, other than Exhibit C, to persons not permitted to view confidential information in this case. For ease of reference, the claim charts attached hereto include all claims previously disclosed by InterTrust, as well as new claims. Numbering/lettering/bolding has been added to the text of each claim for convenience only, and is not intended to alter, expand, or interpret the meaning of those claims. In instances where infringement claims are illustrated by quotation or reference to Microsoft documents, those

22

23

24

25

26

27

28

IT00038608-43419.

(b) Documents re conception, reduction to practice, and/or design/development

InterTrust has produced nonprivileged documents concerning the conception, design, development, and reduction to practice of the inventions disclosed in the patents-in-suit. See, e.g., 1T00000005-17261, 1T00036207-38606, IT00041497-549. In addition, InterTrust has produced voluminous archives of source code created in the course of its business, some of which may constitute additional evidence of the conception, design, development, and reduction to practice of its patented inventions. InterTrust is not currently aware of any other such nonprivileged documents in its possession or control other than said source code and the source code and documents that have been produced.

### (c) Prosecution history of patents-in-suit

The prosecution histories of the patents-in-suit have previously been produced. See, e.g., IT00062350-67643, IT00070342-72434, FH00107455 - 107731, FH00113539-118857,

FH118866-121322.

Dated: September , 2003

KEKER & VAN NEST.

By:

Attorneys for Plaintiff/and Counter-Defendant TERTRUST TECHNOLOGIES

RPORATION

references are intended to be exemplary only, and not limiting.



### **Microsoft Accused Products**

Visual Studio .Net Enterprise Architect Visual Studio .NET Enterprise Developer Visual Studio .NET Professional Visual Studio .Net

ASP.Net

.NET Framework SDK

.Net License Compiler

Office XP Standard

Office XP Professional

Office XP Professional with FrontPage

Office XP Developer

Windows XP Home Edition

Windows XP Professional

Access 2002

Excel 2002

FrontPage 2002

Outlook® 2002

PowerPoint ® 2002

Project 2002

Publisher ® 2002

Visio® 2002

Word 2002

Visio Enterprise Network Tools

Office 2000 SR-1

Project 2000 SR-1

Windows XP Embedded

Windows CE .NET

Windows CE for Automotive

Mobility and Wireless Solutions for business

Mobile Devices

Pocket PC

Microsoft Smartphone Platform

Microsoft XBCX

Windows ME

Digital Asset Server

Microsoft Reader

Windows Media Player

Windows Media Rights Manager SDK

Windows Media Device DRM technology

Microsoft Secure Audio Path technology

Exhibit A

Microsoft System Management Server Windows File Protection System Microsoft ActiveX technology, including all Microsoft tools that support the Microsoft ActiveX licensing model

All products that contain the Microsoft Common Language Runtime (CLR), Microsoft Compact CLR, or Microsoft implemented .Net Common Language Infrastructure

Application Center
BizTalk Server
Commerce Server
Content Management Server
Exchange Server
Host Integration Server
Internet Security and Acceleration Server
Mobile Information Server
SharePoint Portal Server
SQL Server
Windows 2000 Server
.NET Enterprise Services
.NET Infrastructure and Services

Microsoft Installer SDK All products that contain the Microsoft Installer Technology

Microsoft .Net MyServices Windows Hardware Quality Labs Certification Services

Office 2003 and included applications

Server 2003, including Microsoft hosted RMS Services using Passport



3	 	A CELATIMO FINERINGEMENTA
4 5	155.	Products infringing: Any product using Microsoft Product Activation or Reader Activation feature.
3	A virtual distribution environment comprising	
6 7	(a) a first host processing environment comprising	computer running a Microsoft product containing the Product Activation feature, including Windows XP, Office XP, Visio
8	(1) a central processing unit;	2002. Reader using its activation feature.  CPU of computer
9	(2) main memory operatively connected to said central processing unit;	main memory of computer
10	(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer
1	(b) said mass storage storing tamper resistant software designed to be loaded into said main	Microsoft Product Activation software
2	memory and executed by said central processing unit, said tamper resistant software	
3	comprising: (1) machine check programming which	Product Activation software generates
4	derives information from one or more aspects of said host processing	hardware information relating to the host processing environment as part of the
5 .	environment. (2) one or more storage locations	activation process hardware information is stored in the
6	storing said information;	computer's storage
7	(3) integrity programming which (i) causes said machine check	each time the Microsoft program starts up after
8	programming to derive said information,	initial activation, Product Activation checks the originally derived hardware information against current hardware
9	(ii) compares said information to information previously stored	each time the Microsoft program starts up after initial activation, Product Activation checks
	in said one or more storage locations, and	the originally derived hardware information against current hardware
2	(iii) generates an indication based on the result of said	Product Activation software indicates whether the test has passed or failed
3	comparison; and  (4) programming which takes one or more actions based on the state of said indication;	
5	(i) said one or more actions including at least temporarily halting further processing.	Product Activation software will allow system startup procedures to continue, if test succeeds, or discontinue startup and offer user opportunity to reactivate if the test fails
٠.	·	·

Exhibit B

293482.02

27

Product Infringing: Any product using Microsoft Product Activation or Reader Activation feature.  comprising ent computer running a Microsoft product containing the Product Activation feature, including Windows XP, Office XP, Visio 2002
ent computer running a Microsoft product containing the Product Activation feature,
ent computer running a Microsoft product containing the Product Activation feature,
and Reader
y connected main memory of computer it;
connected hard disk or other mass storage contained in computer
r resistant into said said central istant
hing which e or more hardware information relating to the host processing environment as part of the activation process
ions hardware information is stored in the computer's storage
hich
each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information against current hardware
ormation each time the Microsoft program starts up after initial activation, Product Activation checks the originally derived hardware information
cation Product Activation software indicates whether said the test has passed or failed
s one or ate of said
ctions porarily software from generating new files or running user applications if the test fails
n / n _ ell / s _ n iii _ ut _ viiii _ f ost _ ii f _ ela _ a n

Exhibit B

293482.02

27

28

2

	•	• .		
	157.	Product Infringing: Any product using Microsoft Product Activation or Reader Activation feature.		
#	A signal distribution assistant and accomplaint	Activation feature.		
$\mathbb{H}$	A virtual distribution environment comprising	30000000000000000000000000000000000000		
'	(a) a first host processing environment comprising	computer running a Microsoft product containing the Product Activation feature,		
		including Windows XP, Office XP, Visio 200 and Reader		
忊	(1) a central processing unit;	CPU of computer		
r	(2) main memory operatively connected	main memory of computer		
L	to said central processing unit;	·		
	(3) mass storage operatively connected	hard disk or other mass storage contained in		
	to said central processing unit and said main memory;	computer		
(	(b) said mass storage storing tamper resistant software designed to be loaded into said	Microsoft Product Activation software		
	main memory and executed by said central			
	processing unit, said tamper resistant	·		
L	software comprising:	<u>, , , , , , , , , , , , , , , , , , , </u>		
Γ	(1) machine check programming which	Product Activation software generates hash		
l	derives information from one or more	information relating to the host processing		
1	aspects of said host processing	environment as part of the activation process		
L	environment,			
	(2) one or more storage locations	hardware information is stored in the		
⊩	storing said information;	computer's storage		
H	(3) integrity programming which			
	(i) causes said machine check	each time the Microsoft program starts up after		
l	programming to derive said information,	initial activation, Product Activation checks the originally derived hardware information		
		against current hardware		
Γ	(ii) compares said information	each time the Microsoft program starts up after		
1	to information previously stored	initial activation, Product Activation checks		
	in said one or more storage	the originally derived hardware information		
L	locations, and	against current hardware		
ĺ	(iii) generates an indication	Product Activation software indicates whether		
	based on the result of said	the test has passed or failed		
-	comparison; and	<u> </u>		
	(4) programming which takes one or			
-	more actions based on the state of said			
$\vdash$	indication;	Deduct Addition to Access 11 1		
	(i) said one or more actions including displaying a message	Product Activation software displays a message to the user if the test fails		
•	inculand displaying a massage i	THECCAUP IN THE NICET IT THE TEST TOTIC		

Exhibit B | 3

293482.02

27

28

1

ed disk cution.
ed disk
disk
disk
the
n the ensed (aka,
ests an
id to a
ocated
ient VM ring the
st the ient is
<u> </u>
1 100 0
l via a e a urity
· · · · · ·
en lization
en lization

Exhibit B

293482.02

28

1

2

3	FUR U.S. PATENT NO. 5,892,900					
4	157. A virtual distribution environment comprising	Infringing products include: Windows Media Player				
5	a first host processing environment comprising	See 156				
	a central processing unit	See 156				
6	main memory operatively connected to said central processing unit	See 156				
7	mass storage operatively connected to said central processing unit and said main memory	See 156				
8	said mass storage storing tamper resistant software designed to be loaded into said main	See 156				
9	memory and executed by said central processing unit, said tamper resistant software					
10	comprising: machine check programming which derives	See 156				
11	information from one or more aspects of said host processing environment,					
12	one or more storage locations storing said information	See 156				
13	integrity programming which causes said machine check programming to derive said	See 156				
14 15	information compares said information to information previously stored in said one or more storage locations, and					
16	generates an indication based on the result of said comparison; and	See 156				
17	programming which takes one or more actions based on the state of said indication	See 156				
18	said one or more actions including displaying a message to the user.	If these are not equal, the user is notified via a message stating that he/she must acquire a				
19	message to the aber.	security upgrade (that is, the current security upgrade is invalid).				
20						
21						
22						
23	·					
24						
25						

Exhibit

293482.02

26

27

28

3	FOR 0.5.1 ATENT NO. 5,652,500				
_	WEST AND THE THE PROPERTY OF T	ESTREME OF ATMIOFAINTRINGEMENT AND THE PROPERTY OF THE PROPERT			
5	157.	Infringing Product: Microsoft's Windows File Protection and System File Checker features, embodied in Microsoft's Windows 2000, Windows XP products, and Server 2003			
6	A virtual distribution environment comprising				
7	(a) a first host processing environment comprising	computer running Microsoft Windows 2000 or Windows XP.			
8	, <b>,</b>				
9	(1) a central processing unit;	CPU of computer			
10	(2) main memory operatively connected to said central processing unit;	main memory of computer			
11	(3) mass storage operatively connected to said central processing unit and said main memory;	hard disk or other mass storage contained in computer			
12	(b) said mass storage storing tamper resistant software designed to be loaded into said	Windows File Protection process/service ("WFP") and System File Checker (SFC.exe)			
13 14	main memory and executed by said central processing unit, said tamper resistant	features of winlogon.exe. Winlogon.exe is treated as a "critical" service by the Windows			
15	software comprising:	operating system. Files supporting WFP (including winlogon.exe, sfc.exe, sfc.dll (2000 only), sfcfiles.dll (2000 only) and sfc_os.dll			
16		(XP only)) are "protected" files and are signed using a signature verified by a hidden key. In			
17		Windows 2000, WFP uses hidden functions within the sfc.dll library. Functions are imported by "ordinal" instead of "name."			
18 19	(1) machine check programming which derives information from one or more	Winlogon either directly or using another dll (XP) or using SFC.dll (2000) determines if			
20	aspects of said host processing environment,	changed file was protected, computes the hash of protected files and, if necessary, computes			
21		the hash of the file in the dll cache before using it to replace a file overwritten by an incorrect version of the file.			
22	(2) one or more storage locations storing said information;	hardware information is stored in the computer's memory			
23	(3) integrity programming which				
24	(i) causes said machine check programming to derive said information,	Windows notifies Winlogon when there has been a system directory change or a change in the dll cache.			
25		·			
26	(ii) compares said information to information previously stored	Winlogon either directly or using another dll (XP) or using SFC.dll (2000) compares			
27	in said one or more storage locations, and	created from the Catalog file(s), and, if there is			
28		a difference, compares the hash of the file in the dll cache to the hash database created from			

Exhibit B 6 . .

1

		(iii) gener	ates an in	dication		overwr	itten file	; <u>.                                    </u>	re using i	•	
		based on t	he result on; and	of said		An event is written to the Event Viewer if hashes do not agree.					
•	(4) programming which takes one or more actions based on the state of said indication;		Depending on the circumstances, WFP displays several messages to the user, including prompting the user to contact the system administrator, and to insert a CD-RON								
·		(i) said on including to the user	displayin	e actions g a mess	sage	See abo	ove. Me Property	ssages a	lso const	itute vi	ewal
									•	•	
				••	•		• •		•		
						•			•		
			•			•			•		
						•					
•									•		•
									•		
				• .							
					•		٠				
	•		·.								
	,			•							
			•								
		•		٠							
				·							
								•			

293482.02

Exhibit B

4	PAR REPORTED ANGUAGE SERVICE	CEST CEATMORING PINGEMENTS AND AND ADDRESS OF THE PROPERTY OF
5	6.	Product Infringing: XBox
-	A process comprising the following steps:	The process constitutes assembly and use of components making up an XBox game.
6	accessing a first record containing	The first record consists of the second file
7	information directly or indirectly identifying one or more elements of a first	table on an XBox DVD. This table identifies the .xbe file which includes the
.8	component assembly,	game information.
9	at least one of said elements including at	The xbe file includes executable
10	least some executable programming,	programming.
11	at least one of said elements constituting a	The xbe file is a load module.
12	load module,	The xbe file includes a header.
13	said load module including executable programming and a header;	
14	at least a portion of said header is a public portion which is characterized by a relatively lower level of security	Most information the xbe header is not obfuscated.
15	protection; and	
16	at least a portion of said header is a private portion which is characterized, at least	The entry point address and the kernel image thunk address listed in the xbe header are obfuscated and therefore at a
17	some of the time, by a level of security protection which is relatively higher than said relatively lower level of security	higher level of security protection.
18	protection,	
19	using said information to identify and locate said one or more elements;	The second file table identifies the .xbe file, including where that file is located.
20	accessing said located one or more elements;	The .xbe file is accessed by the XBox.
21	securely assembling said one or more elements to form at least a portion of said	At runtime, the .xbe file is assembled with
22	first component assembly;	certain services of the operating system to form a component assembly. Security
23	·	associated with this assembling process includes verifying signatures associated
24		with portions of the .xbe file, and replacing obfuscated calls to operating system
25		services with actual addresses.
26		The assembly may also include patch files downloaded from a remote server.
27		
28	executing at least some of said executable	Game play requires execution of the
	1	• # _ •

Exhibit B

293482.02

1

2

•		•
1	programming; and	assembled programming.
2	checking said record for validity prior to performing said executing step.	The second file table is protected by a digital signature, and is not loaded/used
3		unless the digital signature is verified against the file.
4		
5	7. A process as in claim 6 in which:	
	said relatively lower level of security	The header is protected by the techniques
6	protection comprises storing said public	protecting the xbe such as signing and
_	header portion in an unencrypted state; and	security descriptors, but it is not encrypted except as noted below.
7	said relatively higher level of security	The entry point address and the kernel
8	protection comprises storing said private	image thunk address listed in the xbe
	header portion in an encrypted state.	header are obfuscated. The Xbox SDK's
9		(XDK) image build uses a key value shared
		with the retail XBox to perform two XOR operations against the addresses
10		1 Operations against the addresses
11		
		•
12		
13		
	· ·	
14		
15		
16	·	

Exhibit E

293482.02

STATES AT THE STATE OF A THE STATE OF THE ST	<b>松東於途CEATMORINFRINGEMENTAND</b>
8.	Infringing products: Microsoft CLR or CCLR and .NET Framework SDK and products that include one or both of these.
A process comprising the following steps:	
(a) accessing a first record containing information directly or indirectly identifying	The first record is either an assembly manifest or a whole assembly; the elements are other
one or more elements of a first component assembly,	assemblies that are referenced as external in the first record; the first component assembly is a .NET application domain.
(1) at least one of said elements	Assembly contains executable programming.
including at least some executable programming,	Assembly contains exceeded programmes.
(2) at least one of said elements constituting a load module,	This is an external assembly referenced in the first record.
(i) said load module including executable programming and a header;	Assemblies include executable programming and the assembly manifest and CLS type metadata constitute a header.
(ii) said header including an	This feature is provided for in the .NET
execution space identifier identifying at least one aspect of	architecture through numerous mechanisms,
an execution space required for use and/or execution of the load	permissions.
module associated with said header;	
(iii) said execution space identifier provides the capability	SecurityZone or other evidence provides this capability.
for distinguishing between execution spaces providing a higher level of security and	
execution spaces providing a lower level of security;	
(b) using said information to identify and locate said one or more elements;	Manifest and type metadata information section is used to identify and locate files, coelements, resource elements, individual classes
	and methods.
(c) accessing said located one or more elements;	Step carried out by the CLR or CCLR loader.
(d) securely assembling said one or more elements to form at least a portion of said first	CLR or CCLR carries out this step, including checking the integrity of the load module,
component assembly;	checking the load module's permissions, placing the load module contents into an application domain, isolating it from malicion
	or badly behaved code, and from code that does not have the permission to call it.
(e) executing at least some of said executable programming; and	Step carried out by the CLR/CCLR and the CLR/CCLR host.

Exhibit B

1

2

1	(f) checking said record for validity prior to performing said executing step.	The CLR/CCLR checks the authenticity and the integrity of the first .NET assembly.
2	9. A process as in claim 8 in which said	The CLR/CCLR constitutes a secure
3	execution space providing a higher level of security comprises a secure processing	processing environment.
4 ·	environment.	
5	13. A process as in claim 8 further comprising: (a) comparing said execution space identifier	In one example, the
6	against information identifying the execution space in which said executing step is to occur;	ZoneIdentityPermissionAttribute SecurityZone value demanded by control in the assembly
7	and	manifest is compared against the SecurityZone attribute value corresponding to the calling method
8	(b) taking an action if said execution space identifier requires an execution space with a	CLR/CCLR will throw an exception and transfer control to an exception handler in the
9	security level higher than that of the execution space in which said executing step is to occur.	calling routine, or it will shut down the application if there is no such exception
10 11		handler, if the permissions do not include the permissions required by the ZoneIdentityPermissionAttribute. The
		ZoneIdentityPermissions are hierarchical,
12 13	14. A process as in claim 13 in which said action includes terminating said process prior	unless customized.  CLR/CCLR may terminate the process or transfer control to an exception handler that
14	to said executing step.	may itself terminate the process.
15		
16		
17.		
18		
19	·	·
20	·	·
21		
22		·
23	·	
24		
25		
26		
27		
28		
		the state of the s

Z Z KA GIDANNILAN GOA'GE LEGALA	CLAIMOBENERINGENERIE
8.	Products infringing include Windows Installe SDK, and products that include the Windows Installer technology.
A process comprising the following steps:	Scenario 1: use of Windows Installer package (i.e. MSI files) to create Windows Installer-
	enabled applications, such as Office 2000 and used of the WI service to install them.  Scenario 2: software distribution technologie
	that use the Windows Installer OS service for installation, such as Internet Component
•	Download and products like Office Web Components. Either scenario can be used by SMS,
$\cdot$ .	IntelliMirror and third party tools like InstallShield and WISE.
	NT or later operating systems (because they use the subsystem identifier) using cabinet files, .CAB, (because they have
• .	manifest and INF and/or OSD files), and have been signed with a digital signature and
	will be authenticated by Authenticode or WinVerifyTrust API and contain at least one PE (portable executables
(a) accessing a first record containing information directly or indirectly identifying one or more elements of a first component	Scenario 1: First record is the .MSI file that contains information on what goes in the assembly and how to install the assembly.
assembly,	Scenario 2:  A. First record is the cabinet manifest (indirect instructions)
	B. Or, First record can be INF and/or OSI files (direct instructions)
(1) at least one of said elements including at least some executable programming,	Both scenarios: The PE (portable executable in the cabinet file is the executable programming.
(2) at least one of said elements constituting a load module,	Both scenarios: PE is a load module:
(i) said load module including executable programming and a	Both scenarios: The PE has several headers.

293482.02

1

2

1	header;	
2	(ii) said header including an execution space identifier identifying at least one aspect of	Both scenarios: SUBSYTEM is a field in the PE Optional Header that is an execution space
. 4	an execution space required for use and/or execution of the load module associated with said	
	header;	
6 7	(iii) said execution space identifier provides the capability for distinguishing between	Both scenarios: SUBSYSTEM distinguishes between programs that can run in kernel mode and those that can run in user mode. This is a
8	execution spaces providing a higher level of security and execution spaces providing a	key security concept of process separation that was introduced with Windows NT.
9	lower level of security;	The Subsystem field in the PE header is used by the system to indicate whether the
10 11		executable will run within Ring 3 (user mode) or use Ring 0 (native or kernel mode).
12		Anything running in Ring 3 is limited to its own processing space. Executables running in Ring 0 can reach out to other spaces and have
13		security measure built around them.
14	(b) using said information to identify and locate said one or more elements;	Scenario 1: the MSI file identifies and locates the elements
15		Scenario 2: .CAB manifest is used to identify Physical
16	·	location OSD and/or INF is used to identify Logical
17 18		location
19	(c) accessing said located one or more	Scenario 1: Using the MSI file
20	elements;	Scenario 2: Using INF and/or OSD in cabinet file
21		
22	(d) securely assembling said one or more elements to form at least a portion of said first	Both scenarios: Using the Window Installer OS service with various properties and flags on
23	component assembly;	the settings for higher protection.
24	•	Windows Installer has numerous flags that the developer can set to indicate how the assembly
25		will be installed, in what privilege level, with how much user interface, and how much ability
26		the user has to watch or change what is occurring. These controls have been
27 28		strengthened with each release of Windows Installer. Windows Installer 1.1 and later has
۷ ا		the ability to limit the users capabilities during the installation. In a Windows 2000
	Exhi 1	

2		environment and later, using the Group Policy- based Change and Configuration Management, the administrator has the most control
		the authinishator has the most condu
3	,	Fields that can be set by the developer or
4		administrator to control what users can do
5		include the following:  Transformssecure can be set to a value of 1
	·	to inform the installer that transforms are to be cached locally on the user's computer in a
6		location the user does not have write access.
7		(Transforms create custom installations from a basic generic installation, for example to make
8		the Finance versions different from the
		Marketing version or English versions different from Japanese versions.)
9	·	AllowLockdownBrowse and DisableBrowse
10		cen prevent users from browsing to the sources.
11	·	SourceList can be used to specify the only
12		allowable source to be used for the installation of a given component.
12		Environment can be used to specify whether
13	·	the installation can be done while the user is logged on or only when no user is logged on.
14		Security Summary Property conveys whether
15		a package can be opened as read-only or with no restriction.
·		Privileged Property is used by developers of installer packages to make the installation
16		conditional upon system policy, the user being
17		an administrator, or assignment by an administrator.
18		Restricted Public Properties can be set as
	•	variables for an installation. "For managed installations, the package author may need to
19		limit which public properties are passed to the
20		server side and can be changed by a user that is not a system administrator. Some are
21		commonly necessary to maintain a secure
		environment when the installation requires the installer use elevated privileges. "
22		SecureCusiomProperties can be created by the
23	·	author of an installation package to add controls beyond the default list.
24		MsiSetInternalUI specifies the level of user
25		interface from none to full.  A Sequence Table can be used to specify the
		required order of execution for the installation
26	•	process. There are three modes, one of which is the Administrative Installation that is used by
27		the network administrator to assign and install
28		applications.  InstallServicesAction registers a service for
		the system and it can only he used if the user is

1		an administrator or has elevated privileges with permission to install services or that the
2		application is part of a managed installation.
3		DisableMedia system policy disables media sources and disables browsing to media
4		sources. It can be used with DisableBrowse to
_		secure installations version 1.1 that doesn't have some of the other capabilities.
5		AlwaysInstallElevated can be set per user or
6		per machine and is used to install managed applications with elevated privileges.
7		AllowLockdownBrowse, AllowLockdownMedia and
8		AllowLockdownPatch set these capabilities so
		they can only be performed by an administrator during an elevated installation.
9		[See article "HowTo: Configure Windows
10		Installer for Maximum Security (Q247528).
11	·	Windows XP Professional and .NET have the additional capability to set Software Restriction
12		Policies and have these used by Windows
		Installer.
13		In addition, most of the software distribution
14	·	technologies that use Windows Installer also add a layer of their own controls. For example,
15	·	SMS 2.0 enables the administrators to control the installation is optional or required and
16		whether the user can affect the installation
17	(e) executing at least some of said executable	contents/features at all.   Both scenarios: Part of executable is called
	programming; and	during installation in order to do self-
18		registration or perform custom actions. The overall executable is used at runtime.
19	·	
20	(f) checking said record for validity prior to	Scenario 1: Sign the overall package and the
21	performing said executing step.	cabinet files.
		Scenario 2: The cabinet file is signed.
22		For IE with the default security level or higher,
23		the digital signature is verified by Authenticode or a similar utility before the
24		component is allowed to be assembled.
25		J
26		
27		
	·	
28		
		-
	Fuh	ihia ta

Exhibit B

3		INGEMENT CHART
_	FOR U.S. PATER	NT NO. 5,917,912
4 5	35.	Products infringing include all products that host the Microsoft .NET Common Language
		Runtime or Compact Common Language Runtime.
6	A process comprising the following steps:	
7	(a) at a first processing environment receiving a first record from a second processing	Computer running the Microsoft CLR/CCLR receives, for example, a shared assembly
. 8	environment remote from said first processing environment;	header or a complete shared assembly from another computer, for example a server.
9	(1) said first record being received in a secure container;	The shared assembly is cryptographically hashed and signed.
10	(2) said first record containing identification information directly or	The first record is either an assembly manifest, or a whole assembly; the elements are other
11	indirectly identifying one or more elements of a first component	assemblies that are referenced as external in the first record; the first component assembly
12	assembly;	is a .NET application domain.  Assembly contains executable programming.
13	(i) at least one of said elements including at least some executable programming;	
14	(ii) said component assembly allowing access to or use of	The specified information can include any kind of data file, stream, log, environment variables, etc.
15	specified information; (3) said secure container also including	The shared assembly includes at least some
16	a first of said elements:	executable programming.
17	(b) accessing said first record	CLR/CCLR accesses the assembly or assembly header.
18	(c) using said identification information to identify and locate said one or more elements;	Manifest and type metadata information section is used to identify and locate files, code elements, resource elements, individual classes
19		and methods.
20	(1) said locating step including locating a second of said elements at a third	Met by a multifile assembly, with files distributed across a network, or by the second element constituting another referenced
21	processing environment located remotely from said first processing environment and said second	assembly located elsewhere; the CLR/CCLR uses probing to locate and access the file.
22	processing environment;	uses prooms to rocate and access the me.
23	(d) accessing said located one or more elements;	Step carried out by the CLR/CCLR loader.
24	(1) said element accessing step including retrieving said second	Step carried out by the CLR/CCLR loader.
25	element from said third processing environment;	·
26	(e) securely assembling said one or more elements to form at least a portion of said first	CLR/CCLR carries out this step, including checking the integrity of the load module,
27	component assembly specified by said first	checking the load module's permissions, placing the load module contents into an
28	record; and	application domain, isolating it from malicious or badly behaved code, and from code that

Exhibit B

293482.02

1

(f) executing at least some of said executable programming.	does not have the permission to call it.  Step carried out by the CLR/CCLR.
(1) said executing step taking place at said first processing environment.	CLR/CCLR is operating in the first processing environment specified above.
•	
· ·	
·	
·	
	•

293482.02

4		
5	34.	Product Infringing: Microsoft Operating Systems that support device driver signature technology
6 7	A descriptive data structure embodied on a computer-readable medium or other logic	Signature technology
8	device including the following elements:  a representation of the format of data	The driver package's INF is a data
9.	contained in a first rights management data structure	structure. The INF contains multiple types of sections, structured as hierarchy
10		/"branches," that the Windows operating system or its Plug and Play and/or Set-up
11		installation services "branch" through based on the operating system information and device for which a driver is to be
12		installed. The installation services use the "branching" structure (format) to determine
13		what files should be installed. The INF, further provides disk location information
14		and file directory path information for the files identified as necessary as a result of the "branching" process.
15		
16 17		The driver package is a "rights management" data structure based on the fact that it is governed and based on the fact that it processes governed information.
18		Rights Management as Governed Item
19		A driver manufacturer can include rules
20		governing the driver's installation and/or use in the driver's INF file. For example:
21		Security entries specify an access control
22		list for the driver.  Driver developers can specify rules that determine behavior of the driver package
23		based on the user's operating system version, including product type and suite
24		and the device for which the driver is to be installed
25		Rules specifying logging
<b>26</b> 27		Local administrators can establish policy as to what action or notification should occur
28		in the event that a driver being installed is not signed.
		L

1

2

	·	
2 3 4 5	*	The operating system installation services have a ranking criteria it follows when multiple drivers are available for a newly detected device. The criterion is used to determine the driver best suited for ensuring compatibility with the operating system and ensuring functionality of the device.
6		Drivers have been certified to be
7		compatible with specified operating system versions for their respective device classes.  The catalog file protects the integrity of the
8.		driver.
9		Microsoft distributes the Driver Protection List to prevent known bad deriver from being installed.
11	:	Processing Rights Managed Items
	·	Certain drivers (SAP) have been explicitly
12		certified to protect DRM content.
13		MSDN - DRM Overview
14 15		A DRM-compliant driver must prevent unauthorized copying while digital content
16		is being played. In addition, the driver must disable all digital outputs that can transmit the content over a standard interface (such
17		as S/PDIF) through which the decrypted content can be captured.
18	said representation including:	
19	element information contained within	The elements of a driver package include: A driver that is typically a dynamic-link
20	said first rights management data structure; and	library with the .sys filename extension. An INF file containing information that the
21		system Setup components use to install support for the device.
22		A driver catalog file containing the digital signature.
23		One or more optional co-installers which are a Win32® DLL that assists in device
24		installation NT-based operating systems.
25	·	Other files, such as a device installation application, a device icon, and so forth.
26		XP DDK - INF Version Section
27		The LayoutFile entry specifies one or more
		additional system-supplied INF files that contain layout information on the source
28		media required for installing the software

,		
1		described in this INF. All system-supplied
2		INF files specify this entry.
. 3		The CatalogFile entry specifies a catalog (.cat) file to be included on the distribution
4	organization information regarding	media of a device/driver.  Within an INF is a hierarchy with the top
5	the organization of said elements	being a list of manufacturers, and sub-lists
6	within said first rights management data structure; and	of models and at the bottom a list of install information by model.
7		For Windows XP and later versions of NT-
. 8		based operating systems, entries in the  Manufacturer section can be decorated to
		specify operating system versions. The
9		specified versions indicate OS versions with which the specified INF <i>Models</i>
10		sections will be used. If no versions are
11		specified, Setup uses the specified <i>Models</i> section for all versions of all operating
11	·	systems.
12		INF's SourceDisksNames and
13		SourceDisksFiles sections specify
		organization information.  XP DDK Source Media for INFs
14		The methods you should use to specify
15		source media for device files depend on
16		whether your INFs ship separately from the operating system or are included with the
10		operating system.
17		INFs for drivers that are delivered separately from the operating system
18		specify where the files are located using
		SourceDisksNames and SourceDisksFiles
19		sections.  If the files to support the device are
20		included with the operating system, the
		INF must specify a LayoutFile entry in the Version section of the file. Such an entry
21		specifies where the files reside on the
22		operating system media. An INF that
22		specifies a LayoutFile entry must not include SourceDisksNames and
23		SourceDisksFiles sections.
24		XP DDK – INF SourceDisksNames Section
25		A SourceDisksNames section identifies
23		the distribution disks or CD-ROM discs
26		that contain the source files to be transferred to the target machine during
27		installation. Relevant values of an entry in
21		the INF include:
28		diskid Specifies a source disk. disk-description - Describes the contents
. }		i distraction - Describes the contents

_		
1		and/or purpose of the disk identified by
2		diskid. tag-or-cab-file This optional value
3		specifies the name of a tag file or cabinet file
		supplied on the distribution disk, either in the installation root or in the subdirectory
4		specified by path, if any.
5		path This optional value specifies the path to the directory on the distribution
6		disk containing source files. The path is
U		relative to the installation root and is expressed as \dirname I\dirname 2 and so
7	· .	forth.
8		flags For Windows XP and later, setting this to 0x10 forces Setup to use cab-or-tag-
٠ 9		file as a cabinet file name, and to use tag-
9		file as a tag file name. Otherwise, flags is
10		for internal use only.  tag-file For Windows XP and later, if
11	·	flags is set to 0x10, this optional value
••		specifies the name of a tag file supplied on the distribution medium, either in the
12		installation root or in the subdirectory
13		specified by path. The value should specify the file name and extension without path
14		information.  XP DDK INF SourceDisksFiles Section
15		A SourceDisksFiles section names the
13		source files used during installation, identifies the source disks (or CD-ROM
16		discs) that contain those files, and provides
17	· ·	the path to the subdirectories, if any, on the distribution disks containing individual
18	·	files. Relevant values in an entry in the
		INF would include: filename Specifies the name of the file on
19		the source disk.
20	·	diskid Specifies the integer identifying the source disk that contains the file. This
21		value and the initial path to the
		subdir(ectory), if any, containing the named file must be defined in a
22	·	SourceDisksNames section of the same
23		INF.   subdir This optional value specifies the
24		subdirectory (relative to the
		SourceDisksNames path specification, if any) on the source disk where the named
25		file resides.
26	information relating to metadata, said metadata including:	
27	metadata rules used at least in part to	The driver manufacture can specify rules in
28	govern at least one aspect of use and/or display of content stored within a rights	the INF that govern the installation and/or use of the driver. For example, security
20	management data structure,	entries specify an access control list for the
	il	i

driver. Driver developers can specify rules in an INF file that determines behavior of the driver package based on the user's operating system version, including product type and suite. Also, rules related to logging can be specified as mentioned in next claim element.

### For Example - Access Control List Rules

XP DDK - Tightening File-Open
Security in a Device INF File
For Microsoft Windows 2000 and later,
Microsoft tightened file-open security in
the class installer INFs for certain device
classes, including CDROM, DiskDrive,
FDC, FloppyDisk, HDC, and
SCSIAdapter.

If you are unsure whether the class installer for your device has tightened security on file opens, you should tighten security by using the device's INF file to assign a value to the DeviceCharacteristics value name in the registry. Do this within an addregistry-section, which is specified using the INF AddReg directive.

XP-DDK -- INF AddReg Directive

An INF can also contain one or more optional add-registry-section.security sections, each specifying a security descriptor that will be applied to all registry values described within a named add-registry-section.

A Security entry specifies a security descriptor for the device. The security-descriptor-string is a string with tokens to indicate the DACL (D:) security component. A class-installer INF can specify a security descriptor for a device class. A device INF can specify a security descriptor for an individual device, overriding the security for the class. If the class and/or device INF specifies a security-descriptor-string, the PnP Manager propagates the descriptor to all the device objects for a device, including the FDO, filter DOs, and the PDO.

For Example - Operating System Versioning

Operating-System Versioning for Drivers

1		
1		under Windows XP
2		Setup selects the [Models] section to use based on the following rules:
4		If the INF contains [Models] sections for
5		several major or minor operating system version numbers, Setup uses the section
6		with the highest version numbers that are not higher than the operating system
.7		version on which the installation is taking place.
8		If the INF [Models] sections that match the operating system version also include
· 9		product type decorations, product suite decorations, or both, then Setup selects the
10		section that most closely matches the running operating system.
11	said metadata rules including at least one rule specifying that information	The AddService directive can set up event-logging services for drivers.
12	relating to at least one use or display of said content be recorded and/or	INF AddService Directive An AddService directive is used to control
13	reported.	how (and when) the services of particular Windows 2000 or later device's drivers are
14		loaded, any dependencies on other underlying legacy drivers or services, and
15		so forth. Optionally, this directive sets up event-logging services by the
16	·	devices/drivers as well. Relevant sections of the directive's entry
17		include:   event-log-install-section -Optionally
18		references an INF-writer-defined section in which event-logging services for this
19		device (or devices) are set up.   EventLogType Optionally specifies one
20		of System, Security, or Application. If omitted, this defaults to System, which is
21		almost always the appropriate value for the installation of device drivers. For example,
22		an INF would specify Security only if the to-be-installed driver provides its own
23		security support.  EventName Optionally specifies a name
24		to use for the event log. If omitted, this defaults to the given ServiceName.
25		
26	35. A descriptive data structure as in claim	
27	34, in which: said first rights management data structure	The driver package is secured through a
28	comprises a first secure container.	catalog file that is signed by Microsoft's Windows Hardware Quality Lab and

Exhibit B

	contains the hash of each file of the driver' package. The INF identifies the catalog file used to sign the driver package.
36. A descriptive data structure as in claim 35, in which:	
said first secure container comprises:	The first secure container is the driver package secured by a catalog file.
said content; and	The content is the driver and related files within the signed driver package.
rules at least in part governing at least one use of said content.	The rules are within the INF, which is part of the signed driver package.
37. A descriptive data structure as in claim 36, wherein the descriptive data structure is stored in said first secure container.	The INF is stored within the signed driver package.
44. A descriptive data structure as in claim 34, further including:	
a representation of the format of data contained in a second rights management data structure,	The manufacture and models sections in the INF Version section are provided for the possibility of a single INF representing
data sudetime,	the format for multiple drivers.
	Operating system version "decorating" relating the architecture, major and minor
	operating systems versions, product and suit information all relate to the target environment and is used to identify the
	files necessary for the target environment.
	An INF file, such as in the case of operating system targeting, can be used for
	more than one driver package since it can contain more than one catalog file.
	Further an INF can address the drives
acid accord rights management data	necessary for a multi-functional device.  The files of the second data structure would
said second rights management data structure differing in at least one respect from said first rights management data	vary from the files on the first data structure.
structure.	<u> </u>
45. A descriptive data structure as in claim 44, in which:	
said information regarding elements contained within said first rights	INF specify where the driver files are located using the SourceDiskNames and
management data structure includes information relating to the location of at	SourceDiskFiles sections.
least one such element.	
46. A descriptive data structure as in claim 44, further including:	
a first target data block including information relating to a first target	Operating system version "decorating" relating the architecture, major and minor
·	Exhibit B] 24

1 2	environment in which the descriptive data structure may be used.	operating systems versions, product and suit information all relate to the first target
		environment.
3	47. A descriptive data structure as in claim 46, further including:	
4	a second target data block including information relating to a second target	Operating system version decorating will cover multiple operating systems.
5	environment in which the descriptive data structure may be used,	
. 7	said second target environment differing in at least one respect from said first target	This is the reason for version decorating.
·	environment.	
8	48. A descriptive data structure as in claim	
9	46, further including: a source message field containing	The provider entry in the version section of the INF identifies the provider of the INF
10	information at least in part identifying the source for the descriptive data structure.	file. Also, the INF contains a manufacture section.
11		, seenon.
12		
13		
14		• •
15		
16		
17		
18		
19		
20 21		
22		
23		
24		
25	•	
26		A Comment of the Comm
27		
28		
20		i.

3					
أ	THE SECURIMICANGUAGE SAMES STARTS GLATMOFINERINGEMENT SAMES				
4	58.	Product Infringing: Microsoft Reader SDK and Microsoft Digital Asset Server.			
5	A most and of accepting a first garine	Method is carried out by Microsoft's			
6	A method of creating a first secure container, said method including the following steps:	Digital Asset Server and Microsoft's  Litgen tools			
7	(a) accessing a descriptive data structure, said descriptive data structure	opf file describing the file structure of a protected e-book including metadata,			
8	including or addressing	manifest, and "spine" information			
9	<ol> <li>organization information at least in part describing a required or</li> </ol>	Organization information regarding organization of the ebook and the			
10	desired organization of a content section of said first secure container, and	inscription as specified in the manifest and spine information in the .opf file			
11 12	(2) metadata information at least in part specifying at least one step required or desired in creation of said first secure container;	Metadata constitutes rules specifying the degree of security to use and/or XrML rules			
13 14	(b) using said descriptive data structure to organize said first secure container contents	e-book packaging carried out by Microsoft Litgen tool			
15	(c) using said metadata information to at least in part determine specific	Step performed by Digital Asset Server; example of specific information is			
16	information required to be included in said first secure container contents; and	owner/purchaser information required in the inscription process			
17	(d) generating or identifying at least one rule designed to control at least one	Analyzing the metadata and finally packaging the e-book using a particular			
18	aspect of access to or use of at least a portion of said first secure container	security level specified through the metadata			
19	contents.				
[	71. A method as in claim 58, in which:				
20	(a) said specific information required to be included includes information at	Owner purchaser information required in the inscription process; XrML rule			
21	least in part identifying at least one owner or creator of at least a portion of	requiring display of copyright notice			
22	said first secure container contents.	<u> </u>			

### · 2

## 3

# INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,920,861

7		
5	58.	Product Infringing: All products that host the Microsoft Common Language Runtime or Compact Common Language Runtime.
6	A method of creating a first secure	Method is practiced by a user using the
7	container, said method including the following steps;	Common Language Runtime (CLR) or Compact Common Language Runtime
8	Tonowing steps,	(CCLR) to create a dynamic shared assembly or .NET Framework SDK to
		create a shared assembly
9	(a) accessing a descriptive data structure,	.NET framework Assembly class and/or
10	said descriptive data structure including or addressing	AssemblyBuilder class and/or AssemblyInfo file
11	(1) organization information at least	This information is specified in the classes
12	in part describing a required or desired organization of a content	named above and in the AssemblyInfo file.
13	section of said first secure container, and	
נו	(2) metadata information at least in	This information is addressed in the classes
14	part specifying at least one step	and the AssemblyInfo file, e.g., for a shared
	required or desired in creation of	assembly metadata will be specified that
15	said first secure container;	the assembly is to be signed using specified key
16	(b) using said descriptive data structure to	This step is carried out by applications and
ا ۔۔۔	organize said first secure container	tools using the classes and assembly info
17	contents;	file, including CLR (or CCLR) and .NET Framework SDK
18	(c) using said metadata information to at	This step is carried out by applications and
_	least in part determine specific	tools using the assembly info file and
19	information required to be included in	classes that specify the metadata required
20	said first secure container contents;	in the target assembly
20	and	User may specify rules, as specified in the
21	(d) generating or identifying at least one rule designed to control at least one	NET Framework SDK, to be placed in the
~ 1	aspect of access to or use of at least a	assembly manifest including such rules
22	portion of said first secure container	requiring that all code be managed (CLR or
_	contents.	CCLR compliant), "Code Access Security"
23		permissions be supplied for use of code
		supplied in the assembly, etc
24	64. A method as in claim 58, in which:	
25	(a) said creation of said first secure	Can be a server, PC or workstation running
25	container occurs at a first data	CLR (or CCLR) to create a dynamic shared
26	processing arrangement located at a	assembly or .NET Framework SDK to
20	first site; (b) said first data processing arrangement	create a shared assembly) Included in virtually any computer
27	including a communications port; and	meladed in virtually any compater
·	(c) said method further includes:	
28	(1) prior to said step of accessing said	Download of the assemblyinfo file and/or a
	descriptive data structure, said	file containing a class calling the
ĺ		

Exhibit B

•	il .	
. 1	first data processing arrangement	DefineDynamicAssembly methods r download of SDK containing
2	receiving said descriptive data structure from a second data	assemblybuilder class from a second site
3	processing arrangement located at a second site.	
. 4	(d) said receipt occurring through said first	Communications port is normally used for
5	data processing arrangement communications port.	downloading
	67. A method as in claim 64, further comprising:	
6	at said first processing site, receiving said	Download of the AssemblyInfo file and/or
7	metadata through said communications port.	a file containing a class calling the DefineDynamicAssembly methods or
<b>8</b> °.	•	download of SDK containing assemblybuilder class from a second site
٠. 9	68. A method as in claim 67, in which,	
	(a) said metadata is received separately from said descriptive data structure.	Method practiced when metadata names are addressed by the assembly class and a
10	nom sau descripave data sudettue.	template for the AssemblyInfo file, and
11		values corresponding to those names are received through a user interface such as
12		provided by Microsoft Visual Studio or are provided from a separate file
13	71. A method as in claim 58, in which:	
14	(a) said specific information required to be included includes information at	The Assembly class definition includes attributes for company name and trademark
	least in part identifying at least one	information, and these may be required
15		
15	owner or creator of at least a portion of said first secure container contents.	attributes specified in the AssemblyInfo file
15 16	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:	attributes specified in the AssemblyInfo file
	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to	attributes specified in the AssemblyInfo file  The Assembly class definition includes an
16 . 17 . 18	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20 21	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20 21 22 23	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20 21 22 23 24	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20 21 22 23 24 25	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20 21 22 23 24 25 26	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be
16 17 18 19 20 21 22 23 24 25	owner or creator of at least a portion of said first secure container contents.  72. A method as in claim 58, in which:  (a) said specific information required to be included includes a copyright	The Assembly class definition includes an attribute for copyright field that may be

CHADADAN CHARGE	A CONTINUE IN CONTINUE OF THE PROPERTY OF THE
58.	Product Infringing: Microsoft .NET Framework, Visual Studio .NET, and tools
	that include the Assembly Generator tool AL.exe.
A method of creating a first secure container, said method including the	The Assembly Generation tool generates a portable execution file with an assembly manifest from one or more files that are
following steps;	either Microsoft intermediate language (MSIL) modules or resource files. When
	using the tool's signing option, the assembly becomes a secure container.
(a) accessing a descriptive data structure, said descriptive data structure including or addressing	The descriptive data structure is the text file used as input by the Assembly Generation tool.
(1) organization information at least in part describing a required or desired organization of a content section of said first secure	The DDS specifies the <i>link</i> and or <i>embed</i> directives to indicate which source files should be included in the assembly, how the included resource will be tagged, and if the resource will be private. Private
container, and	resources are not visible to other assemblies.  These tags are used to organize the
	assembly into named sections.  Private attributes are used to organize the assembly into both public and private
	sections. (Public sections are the default.)
(2) metadata information at least in part specifying at least one step required or desired in creation of said first secure container;	The text file can contain "options" relating to how the assembly should be built and additional information that should be included.
	Main - Specifies the method to use as an entry point when converting a
	module to an executable file.  Algid – Specifies an algorithm to hash all files.
· · · · · · · · · · · · · · · · · · ·	Comp – Specifies string for the Company field.
	Conf – Specifies string for Configuration field
	Copy - Specifies string for Copyright field.
	Culture - Specifies the culture string to associate with the assembly.
	Delay - Variation of this option specifies whether the assembly will be

Exhibit B

1	1	_		
2				fully or partially signed and whether the public key is placed in the assembly.  Description - Specifies the description
. 3				Description - Specifies the description field.
3				Evidence - Embeds file in the assembly
4	#	1.	Anna Anna Anna Anna Anna Anna Anna Anna	with the resource name
_				Security Evidence.
5				Fileversion – Specifies the file version of the assembly.
6			•	Flags - Specifies flags for such things
·		1		as the assembly is side-by-side
7	.	1		compatible, assembly cannot execute
_	∦'			with other versions if either they are
8	1		•	executing in the same application domain, process or computer.
٠ 9		-	•	Keyf - Specifies a file that contains a
_			•	key or key pair to sign an assembly.
10	1	1:		Keyn - Specifies the container that holds
		'	•	a key pair.
11				Product - Specifies string for Product field.
12				Producty - Specifies string for Product
		1		Version.
13.		1		Template – Specifies the assembly fro
11				which to inherit all assembly metadata.  Title - Specifies string for Title field.
14		[		Trade - Specifics string for Trademark
15		1		field.
				V - Specifies version information.
16		(b)	using said descriptive data structure to	The following directives are used to specify which files are to be compiled into the
17			organize said first secure container contents	assembly, how they will be tagged, and
1,			contents	whether or not they will be visible to other
18				assemblies, AKA private:
19				Embed[name, private] - copies the
17		1		content of the file into the assembly and
20				applies an optional name tag, and
			·	optional private attribute.
21				Link[name, private] - file becomes part of the assembly via a link and applies an
22				optional name tag, and optional private
<b>LL</b>	.			attribute.
23		(c)	using said metadata information to at	The following are some of the "options"
<b>.</b> .			least in part determine specific	address what information should be
24			information required to be included in	included in the secure container:
25			said first secure container contents;	Main - Specifies the method to use as
			ard .	an entry point when converting a
26-			·	module to an executable file.
_				Comp - Specifies string for the
27				Company field.  Conf - Specifies string for
28				Configuration field
	l			Capy - Specifies string for Copyright
ĺ	<b>I</b> .	e		

	•	· · · · · · · · · · · · · · · · · · ·
		field.  Culture – Specifies the culture string to
		associate with the assembly.
		Description - Specifies the description field.
		Evidence - Embeds file in the assembly
		with the resource name Security. Evidence.
		Fileversion - Specifies the file version
		of the assembly.  Flags - Specifies flags for such things
		as the assembly is side-by-side
		compatible, assembly cannot execute
		with other versions if either they are
	· ·	executing in the same application
		domain, process or computer.
		Keyf - Specifies a file that contains a
•		key or key pair to sign an assembly.
		Keyn - Specifies the container that hold
		a key pair.  Product - Specifies string for Product
		field.
		Producty - Specifies string for Product
	· •	Version.
		Template - Specifies the assembly fro
	·	which to inherit all assembly metadata.
		Title - Specifies string for Title field.
		Trade - Specifics string for Trademark
		field.
		V - Specifies version information. User may specify rules, as specified in the
(d)	generating or identifying at least one	.NET Framework SDK, to be placed in the
` `	rule designed to control at least one	assembly manifest including such rules
	aspect of access to or use of at least a	requiring that all code be managed (CLR
	portion of said first secure container	compliant), "Code Access Security"
	contents.	permissions be supplied for use of code
	·	supplied in the assembly, etc.
71.	A method as in claim 58, in which:	
		The following "options" specifies owner
(a)	said specific information required to	and creator information:
	be included includes information at	
	1	
	least in part identifying at least one	Comp - Specifies string for the
	owner or creator of at least a portion of	Company field.
	least in part identifying at least one owner or creator of at least a portion of said first secure container contents.	Comp – Specifies string for the Company field. Copy – Specifies string for Copyright
	owner or creator of at least a portion of	Company field.  Copy - Specifies string for Copyright field.
	owner or creator of at least a portion of	Company field.  Copy - Specifies string for Copyright field.  Trade - Specifics string for Trademark
	owner or creator of at least a portion of said first secure container contents.	Company field.  Copy - Specifies string for Copyright field.
72.	owner or creator of at least a portion of	Company field.  Copy - Specifies string for Copyright field.  Trade - Specifics string for Trademark field.
	owner or creator of at least a portion of said first secure container contents.  A method as in claim 58, in which:  said specific information required to	Company field.  Copy - Specifies string for Copyright field.  Trade - Specifics string for Trademark field.  The copy "option" specifies the string for
	owner or creator of at least a portion of said first secure container contents.  A method as in claim 58, in which:	Company field.  Copy - Specifies string for Copyright field.  Trade - Specifics string for Trademark field.

	FOR U.S. PATE	N1 NO. 5,982,891
3	II ZAZZE ZE GLAIMILANGUAGEZE ZEGEZ	CHAIMIOFINFRINGEMENT
4	1.	Products infringing: All products that include
_		the Common Language Runtime or Compact Common Language Runtime or Common
5		Language Infrastructure.
6	A method for using at least one resource	Resource may constitute a Microsoft Windows
7	processed in a secure operating environment at a first appliance, said method comprising:	process or hardware element; secure operating environment is Microsoft Common Language
•	<b>2</b>	Runtime ("CLR") environment, Common
.8		Language Infrastructure ("CLI") or Compact CLR ("CCLR"); first appliance is computer
9	·	running CLR, CLI or Compact CLR. Two
		infringing scenarios are set forth herein: (1) For CLR, an administrator, using the .NET
10		framework caspol.exe tool remotely configures
11		security policy in a .NET configuration file for
12	·	a machine, enterprise, user, or application and that security policy interacts with rules or
12	·	evidence declared in a shared assembly
13		provided by another entity ("1" scenario"); and (2) for CLR, CLI and CCLR two assemblies
14		are delivered to an appliance; the first
i		assembly has a rule that demands permissions from a caller in the second assembly, and the
15		second assembly includes a control that asserts
16		such permissions or provides evidence that
17		convinces the runtime that it has such permissions. ("2" scenario"). In each scenario
		Microsoft .NET "Code Access Security"
18		framework or "Role Based Security" framework is used.
19		
20	(a) securely receiving a first entity's control at said first appliance, said first entity being	1 <sup>st</sup> scenario: first entity is the administrator, and the policy that constitutes this entity's
20	located remotely from said operating	control is securely received at the first
21	environment and said first appliance;	appliance through a session established
22		between the administrator's computer and the first appliance, requiring security credentials
	·	such as the administrator's login and password
23		or other secure session means.  2 <sup>nd</sup> scenario: first entity is creator or distributor
24		of the first assembly, assembly manifest
	·	includes a control demanding or refusing or otherwise asserting a security action on
25		permissions from a caller; first assembly is
26.	, , , , , , , , , , , , , , , , , , , ,	integrity-checked.
27	(b) securely receiving a second entity's control at said first appliance, said second entity being	Second entity's control is contained in shared assembly manifest (and therefore integrity
21	located remotely from said operating	protected) that provides evidence for obtaining
28	environment and said first appliance, said	permissions, or asserts permissions; assembly
	second entity being different from said first	creator/distributor is located remotely and is
l		·

Exhibit B

293482.02

2	entity; and	not the administrator (1 <sup>st</sup> scenario) or creator/distributor of the first container (2 <sup>nd</sup> scenario);
3	(c) securely processing a data item at said first appliance, using at least one resource,	Secure processing is carried out by CLR, CLI or CCLR, Data item constitutes an executable
4	including securely applying, at said first appliance through use of said at least one	code element, an interface controlled by such an executable, a data collection or stream (such as media file or stream or text file) or an
5	resource said first entity's control and said second entity's control to govern use of said data item.	environment variable. CLR, CLI or CCLR securely processes the rules, which will in both
7	data nom.	scenarios govern access to methods and data from the first assembly. The resource named in
8		the claim is, e.g., a Windows process that is established by the runtime or hardware element
9	51. A method as in claim 1 wherein at least said secure processing step is performed at an	on the computer.  Consumer computer or appliance running Microsoft CLR, CLI or CCLR).
10	end user electronic appliance.	
11	58. A method as in claim 1 wherein the step of securely receiving a first entity's control	1 <sup>st</sup> scenario 1: link is LAN or WAN; 2 <sup>nd</sup> scenario: link is any telecommunications link,
12	comprises securely receiving said first entity's control from a remote location over a	including the internet.
13	telecommunications link, and the step of securely receiving said second entity's control comprises securely receiving said second	
14 15	entity's control from the same or different remote location over the same or different	
16	telecommunications link.	· · · · · · · · · · · · · · · · · · ·
17	65. A method as in claim 1 wherein the processing step includes processing said first and second controls within the same secure	Secure processing environment is CLR, CLI or CCLR running on user's computer or appliance.
18	processing environment.	In scenario 2, arrangement consists of the stack
19 20	71. A method as in claim 1 further including the step of securely combining said first entity's control and said second entity's control	frame, and the corresponding array of permission grants for assemblies on the stack,
21	to provide a combined control arrangement.	and the permission demanded by the first assembly. Secure combining performed by the CLR, CLI or CCLR,
22	76. A method as in claim 1 wherein said two securely receiving steps are independently	Steps are performed at different times in both scenarios.
23	84. A method as in claim 1 wherein at least one	In both scenarios the second entity supplies an assembly with a demand procedure executed
24 25	of the first entity's control and the second entity's control comprises at least one executable component and at least one data	by the CLR, CLI or CCLR. The data component is a specific attribute value
26	component.  89. A method as in claim 1 wherein said first	referenced by the assembly. Microsoft Common Language Runtime (CLR),
27	appliance includes a protected processing environment, and wherein:	Common Language Infrastructure (CLI), or Compact Common Language Runtime (CCLR)
28	(a) said method further comprises a step of	Typically occurs in both scenarios.
	receiving, at said first appliance, said data item	<u> </u>

	separately and at a different time from said receiving said first entity's control; and														
I	separately and at a different time from said receiving said first entity's control; and  (b) said securely processing step is performed at least in part in said protected processing environment						Protec	cted proc	essing	enviro	ımeni	is th	ne CLF		
i	at least	ın p mer	art in it	said j	NOICE	teu pi	ocessi	ıng	CETO	CCLR.					
		<del>- ; -</del>	<u>·:</u>	<u> </u>				••	<u> </u>				•		
						: ,			· . : .	`. •			٠, •	•	
						·	·							:	
										•			. ;	•	,
									•				•		, · ·
									•	• •					
											٠.		,		
							•	•	N 11. T				• •		
											•				
			•					<i>,</i>		•		٠.			
							·			·					
												•			
	•		•					٠							
١								•							•
			-			•		•							
					-	•									
	٠									•					
١		•										•			
									•		•				
									•				•		
													٠		
	•														
1						•		•	•	••		•			

293482.02

2	INTERTRUST INFRU FOR U.S. PATER	INGEMENT CHART
1	FOR U.S. PATE	11 110. 5,702,071
3		Infinite and dusts include Office 2003 and
4	<b>22.</b>	Infringing products include Office 2003 and included applications, and Server 2003,
5		including Microsoft hosted RMS Service using
ا د		Passport
6	A method of securely controlling use by a third	A user (third party) accesses an IRM-protected
7	party of at least one protected operation with respect to a data item comprising:	data item governed by IRM controls under tw or more RMS servers. For example, the data
_ ′	respect to a data item comprising.	item may be a IRM-protected document.
8		The IRM controls may be associated with the
9.		data item directly or via a IRM-protected
		container holding the IRM-protected data item,
10		such as an IRM-protected email with the IRM- protected document attached.
11	(a) supplying at least a first control from a first	The user acquires a first use license from a first
12	party to said third party;	RMS server (first party) enabling access to, the IRM-protected data item under the IRM rules
12	·	associated with the first RMS server. For
13		example: (1) the first use license from the first RMS server permits the user to access a IRM-
14		protected document contained within or
		attached to an IRM-protected email; or (2) the
15		first use license from the first RMS server applies a first set of IRM rules to an IRM-
16		protected document.
17	(b) supplying, to said third party, at least a second control from a second party different	The user acquires a second use license from a second RMS server (second party) enabling
	from said first party;	access to the IRM-protected data item under
18	•	the IRM rules associated with the second RMS server. For example: (1) in addition to the
19		user being given access to an IRM-protected
		email based on a first use license, a second
20		RMS server provides a second use license enabling access to the IRM-protected
21		document attached thereto; or (2) the second
22		use license from the second RMS server applies a second set of IRM rules to the IRM-
		protected document.
23	(c) securely combining at said third party's	The first and second use licenses are combined
24	location, said first and second controls to form a control arrangement;	to form a control arrangement that governs -access to the IRM-protected data item.
25	(d) securely requiring use of said control arrangement in order to perform at least one	The combined first and second use licenses govern access to the IRM-protected data item.
26	protected operation using said data item; and	
27	(e) securely performing said at least one	The user performs a protected operation (e.g.,
21	protected operation on behalf of said third party with respect to said data item by at least	read, print, edit) on the IRM-protected data item. The combined first and second use
28	in part employing said control arrangement	licenses are employed to permit the protected
		operation.
	rl	inio Di

Exhibit B

1	23. A method as in claim 22 wherein said data	The data item is encrypted and protected by IRM.
2	item is protected.  39. A method as in claim 22 further including	The first and/or second use license are securely and persistently associated with the IRM-
3	securely and persistently associating at least one of: (a) said first control, (b) said second	protected data item.
4	control, and (c) said control arrangement, with	Ga
5	53. A method as in claim 22 wherein at least two of the recited steps are performed at an end user electronic appliance.	Steps performed at a user's computer or appliance.
6	60. A method as in claim 22 wherein step (a)	The first and second use licenses are received
7	comprises supplying said first control from at least one remote location over a	over a telecommunications link such as a networking or modem/serial interface.
8 .	telecommunications link, and step (b) comprises supplying said second control from	
9	the same or different remote location over the same or different telecommunications link	
10	67. A method as in claim 22 wherein at least step (c) is performed within the same secure	Steps are performed at user's computer or appliance.
11	processing environment at said third party's location.	
12	91. A method as in claim 22 wherein: (a) said method further comprises supplying	The first use license (first control) is received
13·	said data item to said third party separately and at a different time from supplying of said first	at the time that the user accesses the data item, which occurs separately and at a different time
	control to said third party; and	from receipt of the IRM-protected data item
14		itself.
14 15	(b) said securely performing step comprises	itself.  The protected operations require decryption of the protected content, which is done inside the
15 16		itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by
15 16 17 18	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21 22	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21 22	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21 22 23	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21 22 23 24	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21 22 23 24 25	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21 22 23 24 25 26	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-
15 16 17 18 19 20 21 22 23 24 25 26 27	(b) said securely performing step comprises performing said protected operation at least in	itself.  The protected operations require decryption of the protected content, which is done inside the RM lockbox. The RM lockbox is protected by mechanisms such as obfuscation, anti-

Exhibit B

	TON O.S. TA	1911 110. 3,702,071
3	26.	Products infringing: Visual Studio.NET,
		.NET Framework SDK, and all products
4		that include the Common Language
5		Runtime or Compact Common Language
١		Runtime or Common Language
6		Infrastructure.
ľ	A secure method for combining data	
7	items into a composite data item	
- 1	comprising:	A first signed and licensed .NET
8	(a) securely providing, from a first location to a second location, a first data item	component, NET assembly, managed
	having at least a first control associated	control and/or Web control (component) is
9	therewith;	the first data item. The first .NET
10		component developer (first location)
,0		provides the application assembly
11		developer (second location) the first
		component. The first control is the set of declarative statements comprising the
12		LicenseProviderAttribute (alternately
		referred to as license controls).
13	(b) securely providing, from a third	A second signed and licensed component is
14	location to said second location, a second	the second data item. The second
17	data item having at least a second control	component developer (third location)
15	associated therewith;	provides the application assembly
		developer (second location) the second component. The second control is the set
16	·	of declarative statements comprising the
.,		LicenseProviderAttribute.
17	(c) forming, at said second location, a	The application assembly developer will
18	composite of said first and second data	include at least the two components into its
"	items;	assembly.
19	(d) securely combining. at said second	At the second location, the application
	location, said first and second controls to	assembly developer uses the .NET runtime that includes the LicenseManager.
20	form a control arrangement; and	that includes the Licenselvianager.
ر, ا		Whenever a component is instantiated
21		(here, an instance of the first licensed
22		component), the license manager accesses
	·	the proper validation mechanism for the
23		component. The license controls (first
		control) for the runtime license (derived from the design time license) are bound
24		into the header of the .NET application
25		assembly, along with the second control for
25		the second component.
26		•
"		Visual Studio.NET securely handles the
27		creation of runtime license controls.
		Runtime licenses are embedded into (and bound to) the executing application
28		assembly. The license control attribute
		assembly. The ficense conduct and bute

Exhibit B

293482.02

1

_		
. 1		included in the first component is
2		customized in the second location to express and require the runtime license. In
3		a more advanced scenario, the License Complier tool can be used to create a
4		".licenses file" containing licenses for multiple components, including runtime
5		licenses for components and classes created by the license provider. This licenses file
. 6	·	is embedded into the assembly.
7		The third control set comprises the runtime license controls for the first and second
8		components (that had been bound to the assembly), the declarative controls
9		provided by the application assembly developer, and any runtime licenses for
10		other components included by the developer in application assembly. The
11		controls are typically integrated into the header of the .NET application assembly
12	(e) performing at least one operation on	The proper execution of the application
13	said composite of said first and second data items based at least in part on said control	will require that the assembly have run time licenses for the two components.
14	arrangement.	
15	27. A method as in claim 26 wherein said	The set of declarative statements comprising the LicenseProviderAttribute of
16	combining step includes preserving each of said first and second controls in said composite set.	both the first and second components are included in the application assembly.
17		
18	28. A method as in claim 26 wherein said performing step comprises governing the	The application will require the first and second controls to operate properly when it
19	operation on said composite of said first and second data items in accordance with	calls the first and second data items, respectively.
	said first control and said second control.	
20	29. A method as in claim 26 wherein said	Signing the component that has embedded
21	providing step includes ensuring the integrity of said association between said	within it the license control ensures the integrity of the association of the control
22	first controls and said first data item is maintained during at least one of	and data item.
23	transmission, storage and processing of said first data item.	
24	31. A method as in claim 26 wherein said	The component includes the license control
25	providing step comprises codelivering said first data item and said first control.	The component includes the license control and therefore they are codelivered.
26		Each component includes the license
27	40. A method as in claim 26 further including the step of securely ensuring that	Each component includes the license control. Signing the component that has
28	at least one of (a) said first control, (b) said second control, and (c) said control arrangement, is persistently associated with	embedded within it the license control ensures the persistence of the association of the control and data item.
	The post of the po	1

		•
1	at least one of said first and second data	
2	items.	
3	54. A method as in claim 26 wherein at least one of steps (c), (d) and (e) is performed at an end user electronic	At least step (e) is typically performed at an end-user electronic appliance.
	appliance.	L
5	61. A method as in claim 26 wherein step	Microsoft maintains Web sites where a
6	(a) comprises providing said first data item from at least one remote location over a	developer can get components over the Web. These sites include references
7	telecommunications link, and step (b) comprises providing said second data item	whereby a developer may obtain components through their Web connection.
. 8	from the same or different remote location over the same or different	One such site is Internet Explorer Web Control Gallery at
9	telecommunications link.	ie.components.microsoft.com/webcontrols
10	68. A method as in claim 26 wherein step (d) is performed within the same secure processing environment at said second	Typically, step (d) will be performed within the same secure processing environment.
	location.	
12 13	79. A method as in claim 26 wherein steps (a) and (b) are performed at different times.	The application assembly developer will typically acquire components at different
14		times.
15	86. A method as in claim 26 wherein at least one of the first and second controls	The component must include an executable and can include a data items as a EULA,
16	comprises at least one executable component and at least one data component.	readme file or help file.
17	00.11.00.10.10.10.10.10.10.10.10.10.10.1	
18	·	
19		
20		er Britania (n. 1864)
21		
22		
23		• • • • • • • • • • • • • • • • • • • •
24		and the second s
25		•
26		•
27		
28		

3		
4	CONTROL CHAIMILANGUAGE RECENS	IN THE STATE OF THE PROPERTY O
5	35	Infringing products include: Windows Media Player, Individualized DRM Clients
6		and the Secure Audio Path (SAP) technology.
7	A method for using at least one resource processed by a secure operating	
8	securely receiving a first load module	The Individualized DRM Client (first load module) is a signed security upgrade DLL.
9	provided by a first entity external to said operating environment	It is also bound to the hardware ID of the machine on which it runs. It is therefore
10		securely delivered and integrity protected:
11	securely receiving a second load module provided by a second entity external to said	A SAP certified driver is also signed and carries with it a certificate that indicates its
12	operating environment, said second entity being different from said first entity; and	compliance with SAP criteria. If it is delivered to a PC it is secure in the sense
13 -		that it is integrity protected. This driver would not come from the same entity as the Individualization DLL.
14	securely processing, using at least one resource, a data item associated with said	If a WM audio file targeted to the Individualized DRM client carries with it a
15	first and second load modules, including securely applying said first and second load	requirement that SAP be supported to render the WMF contents, the content is
16	modules to manage use of said data item.	processed for playing through a soundcard using the WMP and by applying the DRM
17 18		client - which decrypts the content and negotiates with the DRM kernel processing of the content through a Secure Audio Path
19		that includes the SAP-certified audio driver.
20	56. A method as in claim 35 wherein at	All steps occur at the user's PC that
20 21	least two of the recited steps are performed at an end user electronic appliance.	supports the WMP and DRM client and SAP.
		m D' - IDDI(-l'e-tit
22	63. A method as in claim 35 wherein said first load module receiving step comprises	The Driver and DRM client are received from distinct locations and may be
23	securely receiving said first load module from at least one remote location over at	delivered securely over the Internet. They are delivered securely in that each is
24	least one telecommunications link, and said second load module receiving step	integrity protected.
25	comprises securely receiving said second load module from the same or different remote location over the same or different	·
26	telecommunications link.	
27	70. A method as in claim 35 wherein said	Both load modules are executed on the PC
28	securely processing step comprises securely executing said first and second	within the WMP/DRM Client/SAP environment.
	222-21 212-212-2	i

Exhibit B

293482.02

ì

_		
1	ALCOHOL: STATE OF THE STATE OF	MEANGEAIMIOFINFRINGEMENTS HER
2	load modules within the same secure processing environment.	
3	74. A method as in claim 35 further	Since both the DRM client and the driver:
.4.	including securely combining said first and second load modules to provide a	are DLLs in the same audio rendering chain, they exist as an execution
5	combined executable.	environment.
6.	81. A method as in claim 35 wherein said	The driver and Individualization DLL need
7	securely receiving steps are performed independently at different times.	not be received at the same time.
8	94. A method as in claim 35 wherein said	The Windows Media Player together with
9	secure operating environment includes a protected processing environment, and	the Individualized DRM Client and Secure Audio Path comprise a protected environment for processing protected
10	wherein:	media. The protected Windows Media
11	said method further comprises receiving a data item within said secure operating	Files are received after the load modules have been received and installed (licenses cannot be acquired until load modules are
12	environment;	in place). The processing of the Windows
13	said first load module receiving step is performed separately and at a time different	Media File occurs in the protected environment.
14	from receiving said data item; and	
15	said securely processing step is performed at least in part in said protected processing	to the second
16	environment.	<u>.                                    </u>
17	Examples of SAP-certified drivers include -	as indicated at
18	http://www.microsoft.com/Windows/window	wsmedia/wm//DRW/FAQ.asp#Security/
19	All VIA controllers with AC-97 codecs	
20	All ALI controllers with AC-97 codec	•
21	Intel ICH controllers with AC-97 codecs	
22	Creative Labs SoundBlaster 16/AWE32/AWI	E64/Vibra

22

23

24

25

26

27

28

USB Audio

Yamaha OPL3

Yamaha DS-1

ESS Maestro 2E

Cirrus Logic (Crystal) CS4281

Cirrus Logic (Crystal) CS4280

Cirrus Logic (Crystal) CS4614 / CS4624

	1
	2
	3
	4
	5
	6
	7
	8
	9
	0
1	1
	2
	3
	4
	5
	6
	7
	8
	9
	0
	1
	2
. 2	
	4
	.5
	6
2	

28

293482.02

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	•
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	

All SiS c ntr llers with AC-97 codecs

Ensoniq ES1371/73 and CT5880

Ensoniq ES1370

NeoMagic NM6

SoundBlaster Live!

Aureal 8810

Aureal 8820

Aureal 8830

ESS Maestro

ESS ISA parts

NeoMagic NM5

Conexant Riptide

3	FOR U.S. PATENT NO. 5,982,891	
5	36.	Product Infringing: Any product using Common Language Runtime (CLR), Common Language Infrastructure (CLI), or Compact Common Language Runtime (CCLR)
6 7	A secure operating environment system for managing at least one resource comprising:	Microsoft CLR, CLI or CCLR (operating environment system), managing any of the
8		resources on a typical computer, including memory, files system, communications ports, storage devices, and higher level resources that
9	(a) a communications arrangement	may use any of these or combinations of them.  Communications port and Microsoft Internet
10 11		Protocol stack that may optionally use Secure Socket Layer protocol or IPSEC packet security protocol, supplied with Microsoft Windows.
12 13	(1) that securely receives a first control of a first entity external to said operating environment, and	Rule or evidence contained in the manifest of a shared assembly, distributed by a first entity that can be used by the CLR, CLI or CCLR to determine permissions that may be needed to
14 15		cause operations on a data item or resource controlled by another entity; shared assembly is tamper-protected and may be received using secure SSL or IPSEC protocol.
16 17	(2) securely receives a second control of a second entity external to said operating environment, said second entity being different from said first entity; and	Rule specified in the manifest of a second shared (Tamper protected) assembly, that demands permissions of callers of its methods.
18 19	(b) a protected processing environment, operatively connected to said communications arrangement, that:	CLR, CLI or CCLR, connected to (e.g.) communications port
20	(1) [] securely processes, using at least one resource, a data item logically	CLR, CLI or CCLR uses type safety mechanisms, access controls, integrity
21 22	associated with said first and second controls, and	detection, and separation of domains. Data item may be any data item that is managed by the second assembly, which may be a member
23		of such assembly, and whose state or value may be accessible through an interface to other assemblies, and which is referenced by the first
24		assembly,
25	(2) [] securely applies said first and second controls to manage said resource for controlling use of said data	CLR, CLI or CCLR processes the demand for permissions from the second assembly, collects the evidence or processes the rule from the first
26	item.	assembly, and determines whether the first assembly has the permissions to use the
27		resource to operate on the data item controlled by the second assembly.
28	57. A system as in claim 36 wherein said protected processing environment is part of an	Computer or electronic appliance running CLR, CLI or CCLR

Exhibit B 43

293482.02

1

	·	
1		·
2	end user electronic appliance.  64. A system as in claim 36 wherein said	Shared assemblies are designed to be received
3	communications arrangement receives said first and second controls from at least one remote location over at least one	remotely, e.g., over the internet.
4	telecommunications link.	American anniets of the stack forms and
. 5	75. A system as in claim 36 wherein said protected processing environment combines said first and second controls to provide a	Arrangement consists of the stack frame and and the corresponding array of permission grants for assemblies on the stack, and the
6	combined control arrangement.	permission demanded by the second assembly.
7	82. A system as in claim 36 wherein said communications arrangement independently	Assemblies, including controls, are designed for independent delivery.
8	receives said first and second controls at different times	
9	88. A system as in claim 36 wherein at least one of the first control and second controls	The second entity supplies an assembly with a demand procedure (executed by the CLR, CLI
10	comprises at least one executable component and at least one data component.	or CCLR) that includes reference to a specific attribute value (the data component), and the
11		protected processing environment executes the executable component (demand) in a manner
12		that is at least in part responsive to the data component (execution is in response to the
13		security action supplied in the data item).
14		
15	·	× .
16		
17		
18		
19	·	
20		
20	· .	
21	·	
22		
23		
24		
25	•	•
26	·	
27		·
28		
	·	

4	LESSES ALCHAIMHEANGUAGEAS COM	CEAIMIOF INFRINGEMENTAL
5	36.	Infringing Product: My Services
6	A secure operating environment system for managing at least one resource	Secure operating environment is the secure server for any .NET My Services service
7	comprising: a communications arrangement that	(e.g. My Calendar, My Inbox) Secure server receives communications
8	securely receives	formatted using the SOAP-SEC, the security extension to SOAP that is used by
9	·	My Service servers to receive controls.
10		
11		m 6
12	a first control	The first control is a roleTemplate associated with the service. The roleTemplate identifies specific actions
13		(e.g. read, replace) that can be performed against a certain scope (resource or set of
14		resources).
15		The Court austine is the administrator of the
16	of a first entity external to said operating environment,	The first entity is the administrator of the server database, or other entity with authority over its content that sets up the
17		roleTemplates and scopes. That entity is independent from and located remotely
18	and securely receives a second control	from the secure server.  A role element specified by a specific end
19	and securely receives a second control	user, which is securely received by the secure server using the SOAP-SEC
20		protocol.
21		
22	of a second entity external to said operating environment, said second entity	The end user is located remotely from the secure server.
23	being different from said first entity; and a protected processing environment,	The protected processing environment is
24	operatively connected to said	the .NET security service (authorization
25	communications arrangement, that:	system) operating within the server. The server uses the SOAP-SEC
26		communication protocol to receive controls.
27	(a) securely processes, using at least one resource, a data item logically associated	"Securely processes" is performing the requested operation on secure server
28	with said first and second controls, and	running .NET. The system will perform the requested operation ensuring that the user
		has no access to information outside the
- 1		

Exhibit B

293482.02

1

2

1		scope computed.
3		The resource is the server software and/or hardware used to process the two controls and user data.
·4 5		The first control is the roleTemplate for the service. The second control is the role element for an individual user.
6		The data item is the end user's stored
7	· ·	content (e.g. calendar, email inbox, etc.).
8	(b) securely applies said first and second	The secure server determines the result
9	controls to manage said resource for controlling use of said data item.	scope (visible node set) for the operation that is computed from the role element and the roleTemplate. That result scope is used
10		to manage the data item.
11		
12	64. A system as in claim 36 wherein said	The remote location is the site where the
13	communications arrangement receives said first and second controls from at least one	user's or administrator's application is running.
14 15	remote location over at least one telecommunications link.	The telecommunication link can be the Internet, intranet, VPN or other similar channels.
16	75. A system as in claim 36 wherein said	The role scope incorporating the role
17	protected processing environment combines said first and second controls to	element and the role Template.
18	provide a combined control arrangement.	
19	82. A system as in claim 36 wherein said communications arrangement	Administrator and user controls will ordinarily be received at different times.
20	independently receives said first and second controls at different times.	
21	95. A secure operating environment system	This is the normal case for .NET My
22	as in claim 36 wherein said communications arrangement also receives	Services. The user's content is normally stored and updated independently of the
23	a data item separately and at a different time from at least one of said first control	setting of scope elements, role elements and roleTemplates.
24	and said second control.	
25		
26		·
27		

		· ' '
4.	HERTOLOGICATMITANGUAGESCARA	EAST AND ENGINERING EMENDS
5	COMPANY OF THE PARTY OF THE PAR	Product Infringing: Windows CE for Automotive
ا د	1. A security method comprising:	WCEIA is Microsoft Windows CE for Automotive,
6	·	sometimes also known by its former name, AutoPC 2.0.
7		With WCEfA an OEM can assign their device to a class that only accepts certain kinds of software. The device
8		can be set to accept 1) any software with the correct processor/version 2) only certified software or 3) only
9		software from the OEM or Microsoft. These Security (r Trust) levels also control to which kernel APIs and middleware APIs the software has access.
0	,	Intodiewate At 15 the software has decease
1	-	Background: "Microsoft Software Install Manager (SIM), a
2	·	component of WCEfA, allows you to control what can be installed on your device platform. You can define
3		your platform as being open, closed or restricted to new installations, and SIM will enforce these designations."
4		(D,pg.1)
5		"Anything can be installed on an open platform, as long
		as the applications are compiled for the appropriate processor. At the other extreme, no third-party software
6	·	can be installed on a closed platform. Only certified
7		applications can be installed on a restricted platform." (D, pg. 1)
8		"By restricting installations to compliant applications,
9		the risk of installing and using incompatible or harmful software is greatly reduced, while still keeping the
20		device open for robust, quality applications that enhance
		the user experience." (F, pg.1)
21	· ·	WCEfA also has a Security Layer whose purpose is to "Create an abstraction layer of security surrounding ISV
22		applications to limit and/or deny access to key Windows
23		CE kernel API calls and WCEfA middleware APIs." I, pg. 1)
24		A first load module is a WCEfA software comp nent in
25	(a) digitally signing a first load module with a first digital signature designating the first load	a signed .PE file. The first device class is a device that
	module for use by a first device class;	only allows software designated as "restricted" (or higher) to be installed. "Restricted" software is software
26		that has been certified. With restricted software, the
27		device also implements a Security Layer functionality that limits the kernel and WCEfA API calls that the
28		s ftware can make.
:		

Exhibit B

1

2

_	;
1 ·	"SIM Level: 1 = Restricted
2	Description: Only properly certified CEI (WCEfA device installation) files can be installed on the devic.
3	Remote executi n is restricted to executables with master key.
4	Key: Logo certified CEI file required. CEI files or EXEs with master keys permitted." (F, pg.1)
5	
6	"The kernel loader calls it each time a module is loaded by Windows CE. It returns one of the following values
	that determine the module's access to kernel resources:
7	Value Meaning
8	
9	OEM_CERTIFY_TRUST (2) The module is trusted by the OEM to perform any
10	operation.
11	OEM_CERTIFY_RUN (1) The module is trusted by the OEM to run but is
12	restricted from making certain function calls.
13	OEM_CERTIFY_FALSE (0)
14	The module is not allowed to run.
	" (H, pg. 1)
15	Digitally signing: "Before the kernel loads a file, it uses the OEMCertifyModule function to verify that the file
16	contains the proper signature." (N, pg.1)
17	"Signfile.exe: This tool signs an executable with a supplied private key. You can use the following
18	command parameters with this tools AttribString,
19	specifies an optional attribute string to be included in the signature. For example, you could add a string to
20	indicate the trust level of the application." (O. Pg. 1)
21	In the MSDN article Verifying the Signature, the sample code segment states
22	"//the file has a valid signature  // we expect the trust level to be returned as signed
23	data
24	//case 'R': dwTrustLevel = OEM_CERTIFY_RUN" (N, pg.2)
25	"The WCEfA Security Layer isolates installed applications from making unrestricted kernel and
26	WCEfA API calls. This allows the OEM to assign one of three levels of security to applications and drivers
27	installed in RAM when they are loaded into the system.
28	The three levels are Trusted, Restricted, and BlockedOn the systems level, the WCEfA Security

Exhibit B

	layer fits between ISV applications and isolates these software modules from having free access to all WinCE
	kernel calls and WCEfA middleware APIs." (I, pg. 1)
	The developer submits their application for certificating If it passes, then the .cei file (a form of cab file) receive
	a certification key from the certifier. The signed PE is
	within this .cei file.
(b) digitally signing a second load module with a second digital signature different from the	A second load module is a WCEfA software component is a signed PE file. The second device class with a
first digital signature, the second digital signature designating the second load module	different tamper resistance or security level is a device that is "Closed", that is, it will not allow third party to software to be installed. A closed device only allows
for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper	trusted software to run. The Security Layer setting of "Trusted" allows the Microsoft and OEM software full
resistance and security level of the first device class;	access to kernel and middleware APIs.
•	In the MSDN article <u>Verifying the Signature</u> , the samp code segment states
	"//the file has a valid signature // we expect the trust level to be returned as signed
	data  //case 'T' : dwTrustLevel = OEM_CERTIFY_TRUST'   (N, pg.2)
·	"Signfile.exe: This tool signs an executable with a
·	supplied private key. You can use the following command parameters with this tools AttribString,
-	specifies an optional attribute string to be included in the signature. For example, you could add a string to indicate the trust level of the application. (O. Pg. 1)
	"SIM Level: 2 = Closed
	Description: Platform is limited to software supplied directly by OEM or Microsoft. Third-party application cannot be installed
	Key: Master key required for any install or remote execution." (F, pg.1)
	Related to the Security Layer, the Trusted level "is mo likely reserved for MS and OEM applications and
	drivers." (1, pg. 1)
	Whereas the .cei files for certified software have a certification key (sometimes call MS Logo key), the .c
	files from Microsoft or the OEM have a master key attached. ""Master key required for any install or remove execution." (F, p.g.)
(c) distributing the first load module for use by at least one device in the first device class; and	First load module is the certified software from a third party that will be run as part of the "Restricted" first device class.
	"Once your application is complete, send the .cei file t

1 2 3 4 5 6		the organization that is performing validation or certification for the OEM. They would validate it, then either reject or return a .cei that has been stamped with a certification key. You would then reproduce this .cei file on CD-ROM or a compact flash card and distribute." (D, p.g 5)  "APCLoad compares the device SIM level against the .cei file certification key, and either allows the installation to proceed or prohibits it based on the outcome of this comparison." (D, pg. 2)
7		"Security;. To achieve a high level of reliability,
8		WCEfA is carefully designed to: - Control the installation of certified and tested
. 9		software and drivers.  - Limit the access of system services by installed
10		module.  - Monitor the proper execution of software"
11		(G, pg. 1)
12 13	(d) distributing the second load module for use by at least one device in the second device class.	The second load module is the certified software fr m the OEM or Microsoft that will be run as part of the "Closed" second device class.
14 15 16		"You may need to change ROM components after y ur device ships, either to fix a problem, or to provide enhanced functionality. For this purpose, the OEM is given a CElBuild that adds a master key to a .cei file. CEl files stamped with this master key can be installed on an open, closed or a restricted platform." (D, pg. 3)
17 18 19 20		"Trusted: The application is registered as a completely trusted module and allowed full access to the kernel APIs and WCEfA APIs. This mode is mostly likely reserved for MS and OEM applications and drivers. Note that applications and drivers included in ROM are automatically given trusted status." (1, pg.1)
21 22 23 24 25	References: [D] http://msdn.microsoft.com/library/default.asp?url=/lib [F] http://msdn.microsoft.com/library/default.asp?url=/lib [G] http://msdn.microsoft.com/library/default.asp?url=/lib [H] http://msdn.microsoft.com/library/default.asp?url=/lib [I] http://msdn.microsoft.com/library/default.asp?url=/lib [N] http://msdn.microsoft.com/library/default.asp?url=/lib [O] http://msdn.microsoft.com/library/default.asp?url=/lib	rary/en-us/apcguide/htm/ceibuildrev_8.asp rary/en-us/apcguide/htm/securityrev.asp rary/en-us/apcguide/htm/securityrev_7.asp ary/en-us/apcguide/htm/reliabilityrev_3.asp rary/en-us/wcedsn40/htm/cgconVerifyingSignature.asp
26	·	
ļ	·	·
27	·	
28		
•	Fyhi	ihit B

3		
4 5	5.	Product infringing: Windows Hardware Quality Lab certification services, and
6		operating system products that support driver signature technology.
.7	A software verifying method comprising:	Microsoft encourages manufacturers to have their device drivers tested and signed. For example, only signed drivers will ship "in-the-box." Also, Microsoft's driver
8 9		ranking prefers signed drivers to unsigned drivers.
10		Microsoft Web Page - Can't Find a Test Category for Your Driver?
11	·	WHQL's long-term objective is to be able to digitally sign all drivers. Although we do
12		not currently have test programs for certain driver types, such as specialized device drivers and software filter drivers, WHQL
13		is investigating a long term solution to expand the categories of drivers tested
14 15		under Windows 2000 and ultimately all Windows operating systems. We are
16		already formulating a test program for anti- virus file system filters, and plan to address other file system filter drivers as soon as
17		the initial program is in place.
18	(a) testing a load module	The driver will be tested for each version of the operating system it supports and against the device class specification that apply to
19		the device's class.
20		The driver package is a load module. A driver package contains one or more of the
21 -		following files: A device setup information file (INF file) A driver catalog (.cat) file
22		One or more optional co-installers
23 24		Microsoft operates the Window Hardware Quality Lab, which tests drivers submitted
25		by driver manufactures.
26		The manufacturer can test their own driver using the Microsoft testing kit and submit the test results to WHQL when requesting a
27	•	signature. Additionally, Microsoft or a testing facility working with Microsoft can
28	having at least one anneithering associated	perform the testing.  The manufacturer-written INF file, which
	having at least one specification associated	The manuacturer-withen har me, which
	•	

	•	
1	therewith,	is part of the driver package, is a
2		specification. Microsoft Windows drivers must have an INF file in order to be
. 3		installed.
	the specification describing one or more functions performed by the load module;	The INF Version section specifies its device class. One use of the device class is
4	Tunctions performed by the load modern,	to identify the specific Windows
5		compatibility specification that relate to the device class. These specifications will vary
6		by device class in part because the function of each device can vary among class. The
7		INF incorporates by reference the Microsoft supplied device class-specific
8		specification by identifying its class in the
		INF.
9		The INF can include operating system "decorating" to specify the operating system architecture, major and minor
11		version, product and suite the driver is intended for and can further use this
12		decorating to specify what operating systems for which it is not intended.
-13		Because the functionality of each of the
		operating systems may vary the driver must be tested for each applicable operating
14		system.
15 16		<u>Qualification Service Policy Guide –</u> <u>Hardware Category Policies</u>
		You must select the correct hardware
17	·	category for your device. If you select the wrong hardware category for your device,
18		your submission will fail. For example, if
19		you have a storage/hard drive device, but you select storage/tape drive as your
20		hardware category, your submission will
		fail.
21		Windows XP HCT 10.0 Q & A – Windows   XP Logos
22		
23		Q: Which "Designed for Windows XP" logos are available for my product?  A: Devices and systems qualify for a
24.	and the same of th	A: Devices and systems qualify for a "Designed for Windows" logo after passing
25	·	testing with the appropriate WHQL test kit on all operating systems specified by the
26		logo. "Designed for Windows" Logos for Device
27		and System Programs lists which logos are available for each type of product.
	(b) verifying that the load module satisfies	The Microsoft WindowsXP Hardware Compatibility Test (HCT) kit version 10.0
28	the specification; and	includes the tests, test documentation, and

. 1		•
1 2 3 4 5 6		submission processes that are required to participate in the Microsoft Windows Logo Program for Hardware for the Windows XP Professional operating system. To qualify to use the "Designed for Windows" logo for hardware, products must pass testing with the Microsoft Windows HCT kit. The HCT kits are organized by hardware type.  As mentioned above, the manufacturer can test their own driver using the Microsoft testing kit and submit the test results to
8		WHQL when requesting a signature.  Additionally, Microsoft or a testing facility working with Microsoft can perform the
(c) issuin	g at least one digital certificate	When a driver package passes WHQL
attesting t	to the results of the verifying ste	file containing a hash of the driver binaries and other relevant information. WHQL
		then digitally signs the CAT file using Digital Signature cryptographic technolog and sends it to the vendor. Driver signing
		does not change the driver binaries or the INF file submitted for testing.
	·	Microsoft uses digital signatures for device
	•	drivers to let users know that drivers are compatible with Microsoft Windows XP, Windows 2000, and Windows Me. A driver's digital signature indicates that the
		driver was tested with Windows for compatibility and has not been altered sintesting.
	·	•
	· .	
	· .	
·		

	FUR U.S. PATENT NO. 6,137,721				
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT			
5	14.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport			
7	A first protected processing environment comprising:	A personal computer running Windows XP, Windows 2000, or Windows 2003			
8	a first tamper resistant barrier having a first security level, and	The tamper resistant barrier is the Office 2003 IRM client environment and includes the			
. 9		signed digital certificate identifying the user.			
10		If the certificate is tampered with, or if certain, sensitive IRM processes or modules are			
11		debugged or tampered with, the system will cease to operate.			
12	·	The first security level is the "Security Level"			
.13.		which has been selected for a particular Office Application, e.g., Word.			
14	at least one arrangement within the first tamper resistant barrier that prevents the first	The arrangement that prevents a load module from running in one PPE and not in another is			
15	protected processing environment from executing the same load module accessed by a	the type and characteristics of a particular Load Module (VBA program within a document or			
16	second protected processing environment having a second tamper resistant barrier with a	add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level"			
	second convity level different from the first				
17	second security level different from the first security level.	settings.			
18	second security level different from the first				
18 19	second security level different from the first				
18 19 20	second security level different from the first				
18 19 20 21	second security level different from the first				
18 19 20 21 22	second security level different from the first				
18 19 20 21 22 23	second security level different from the first				
18 19 20 21 22 23 24	second security level different from the first				
18 19 20 21 22 23 24 25	second security level different from the first				
18 19 20 21 22 23 24 25 26	second security level different from the first				
18 19 20 21 22 23 24 25 26 27	second security level different from the first				
18 19 20 21 22 23 24 25 26	second security level different from the first				

Exhibit B

3	FOR U.S. FATEN1 NO. 0,157,721				
	CLAIM LANGUAGE S	CLAIM OF INFRINGEMENT			
4	18.	Infringing products include Office 2003 and			
5		included applications, and Server 2003, including Microsoft hosted RMS Service using			
6		Passport			
7	A method for protecting a first computing arrangement surrounded by a first tamper	The first computing arrangement with a tamper resistant barrier is the Office 2003 IRM client			
8	resistant barrier having a first security level, the method including:	environment and includes the signed digital certificate identifying the user.			
9		If the certificate is tampered with, or if certain,			
10		sensitive IRM processes or modules are debugged or tampered with, the system will			
11		cease to operate.			
12		The computing arrangement is being protected from; for example, viruses and malicious code.			
13		The first security level is the "Security Level"			
14		which has been selected for a particular Office Application, e.g., Word.			
15	preventing the first computing arrangement from using the same software module	The arrangement that prevents a load module			
16	accessible by a second computing arrangement having a second tamper resistant barrier with a	from running in one computing arrangement and not in another is the type and			
17	second security level different from the first security level.	characteristics of a particular software module (VBA program within a document or add-in);			
18		i.e., signed, script author, code capabilities, etc., and the "Security Level" settings.			
19					
20	· .				
21					
22		•			
23					
24		•			
25					
26					
27					
28					

Exhibit B

2	INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,157,721			
3	CLAIM LANGUAGE	SCLAIM OF INFRINGEMENT		
- 4 5 6	34.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport		
7	A protected processing environment comprising:	A personal computer running Windows XP, Windows 2000, or Windows 2003		
8 9 10 11	a first tamper resistant barrier having a first security level,	The first tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital certificate identifying the user. If the certificate is tampered with, or if certain, sensitive IRM processes or modules are debugged or tampered with, the system will cease to operate.		
12		The first security level is the "Security Level" which has been selected for a particular Office Application, e.g., Word.		
13 14	a first secure execution space, and	The secure execution space is process space allocated by the operating system for the Microsoft Office host application to run. This		
15 16		host application (e.g., Word) executes the VBA code within this process space.		
17 18		This execution space (application) is secure because the IRM environment takes steps to insure that it is "trusted", the application is signed, and the document which includes the VBA code is protected by IRM policy and then encrypted and signed.		
19 20	at least one arrangement within the first tamper resistant barrier that prevents the first	The arrangement that prevents a load module		
21	secure execution space from executing the same executable accessed by a second secure execution space having a second tamper	from running in one computing arrangement and not in another is the type and characteristics of a particular software module		
22	resistant barrier with a second security level different from the first security level.	(VBA program within a document or add-in); i.e., signed, script author, code capabilities,		
23		etc., and the "Security Level" settings.		
24	•			
25 26				
27				

Exhibit B

293482.02

28

J		
4	MARKET CLAIM LANGUAGE EXPERIENCE	A CONTROL OF THE RINGEMENT AND A SECOND OF THE RINGEMENT AND A SEC
5	34.	Product Infringing: Microsoft Common Language Runtime and ASP.NET
6	A protected processing environment comprising:	Microsoft Common Language Runtime and ASP.NET
7	a first tamper resistant barrier having a first security level,	TAMPER RESISTANT BARRIER The first tamper resistant barrier is the application
8	mst seedarly level,	domain in the CLR. The runtime hashes the contents of each file loaded into the application
.9		domain and compares it with the hash value in the manifest. If two hashes don't match, the assembly fails to load.[1]
0		
1		Also "Code running in one application cannot directly access code or resources from another application. The common language runtime
2		enforces this isolation by preventing direct calls between objects in different application domains.
3.		Objects that pass between domains are either copied or accessed by proxy."[2]
4		SECURITY LEVELS
5		The security levels of the application domain if
6		different by setting the trust level assigned to an outside application using the "trust" element in the web.config for the ASP.NET application.
.7		Syntax- <trust <="" level="Full/High/Low/None" td=""></trust>
9		originUrl="url"/>
0.		Example- <trust <="" level="High" td=""></trust>
1		onginUrl=http://www.SomeOtherCompany.com/default.aspx/>
22		[7]
3	a first secure execution space, and	The application domain is the execution space for a particular application.
4 ·	at least one arrangement within the first tamper resistant barrier that prevents the	The second secure execution space is another application domain that has a different trust level for
5	first secure execution space from executing the same executable accessed	an outside application.
6	by a second secure execution space having a second tamper resistant barrier	If second app domain gives Full trust to the outside application; whereas the first one doesn't, the first
7	with a second security level different from the first security level.	app domain won't be able to execute the application that requires full trust permission.
28		References:
-		Potentia pil

293482.02

1

2

3

	www.microsoft.com/germany/ms/msdnbiblio/dotnetrk/doc/assembly.doc [2] msdn.Microsoft.com/library/en-us/cpguide/html/cpconapplicationdomainsoverview.asp?frame=tue [7] LaMacchia,etc, .NET Framework Security, Addision-Wesley, 2002
•	

<b>3 ∥</b>	
4 MENANGUNGEN SENERAL	MANAGEAIM OF INFRINGEMENT SEE
34. 5	Product Infringing: Products containing Microsoft Common Language Runtime or Compact Common Language Runtime and products implementing the Common Language Infrastructure specification.
A protected processing environment comprising:	Microsoft Common Language Runtime and .NET Framework SDK:
a first tamper resistant barrier having a first security level,	TAMPER RESISTANT BARRIER The first tamper resistant barrier is the
	application domain in the CLR. The runtime hashes the contents of each file loaded into the
	application domain and compares it with the hash value in the manifest. If two hashes don
	match, the assembly fails to load. [1]
2.	Also "Code running in one application canno directly access code or resources from another
	application. The common language runtime enforces this isolation by preventing direct
	calls between objects in different application domains. Objects that pass between domains
	are either copied or accessed by proxy."[2]
	SECURITY LEVELS
	Application domains have different security levels by setting security policy of the
	application domain programmatically. [3] "It has different security based on code-based security model of .NET. Administrators and
	hosts use code-access security to decide what code can do, based on characteristics of the
	code itself, regardless of what user is executive the code. The code characteristics are called
	evidence and can include the Web site or zone from which the code was downloaded, or the
	digital signature of the vendor who published the code."
	"When the security manager needs to
	determine the set of permissions that an assembly is granted by security policy, it start
	with the enterprise policy level. Supplying the assembly evidence to this policy level will
-	result in the set of permissions granted from that policy level. The security manager
	typically continues to collect the permission sets of the policy levels below the enterprise
	policy [including the app domain] in the same

Exhibit B

1

2

1 2 3 4		fashion. These permission sets are then intersected to generate the policy system permission set for the assembly. All levels must allow a specific permission before it can make it into the granted permission set for the assembly."
5		Example of granted permission sets from a policy — Condition: All code, Permission Set: Nothing
7		Condition: Zone: Internet, Permission Set: Internet Condition: URL: www.monash.edu.au, Permission Set: MonashPSet
		Condition: Strong Name: m-Commerce, Permission Set: m-
. 9		CommercePSet [4]
10 11		Another difference in security levels can be whether the verification process is turned off or on, "Managed code must be passed through a
12	·	verification process before it can be run (unless the administrator has granted
13		permission to skip the verification). The verification process determines whether the
14		code can attempt to access invalid memory addresses or perform some other action that
15		to fail to operate properly. Code that passes
16	·	the verification test is said to be type-safe. The ability to verify code as type-safe enables the
17 18		common language runtime to provide as great a level of isolation as the process boundary, at a much lower performance cost." [5]
	·	
19	a first secure execution space, and	The application domain is the execution space for a particular application.
20	at least one arrangement within the first tamper resistant barrier that prevents the first secure	The second secure execution space is another application domain that has a different security
21	execution space from executing the same	policy than the first.
22	executable accessed by a second secure execution space having a second tamper	If second app domain's security policy doesn't
23	resistant barrier with a second security level different from the first security level.	give any permission to code from internet zone, but first app domain does, then the code
24		would run in first app domain and not in second.[6]
25		References:
26		[1] www.microsoft.com/germany/ms/msdnbibl
27		io/dotnetrk/doc/assembly.doc [2] msdn.Microsoft.com/library/en-
28		us/cpguide/html/ cpconapplicationdomainsoverview.asp?fra me=true

					[3] LaM Security [4] Watl Security MSDN [5] sam [6] msdi us/cpgui cpconap .asp?fra	Macchia, etc., Addision kins, Demir in the .NI Library, Jace as [2] n.Microsolide/html/oplicationd me=true	:, .NET i-Weste ien, "A ET Fran inuary ft.com/	Framey, 200 n Ove mewor 2002 library	ework 12, p.11 rview c k", fro //en-
	•			<u>.</u>				<del>!</del>	<del></del>
				•	,				
	•							ij	•
		•		.··			•	. "	
			<i>:</i>						
			٠						
•				•	•	•			
÷ .					·		•		
	•								
-						•	•		
•									
					. ,		,	•	

293482.02

	FOR U.S. PATENT NO. 0,157,721				
3	CLAIMICANGUAGE	CLAIM OF INFRINGEMENT			
4	38.	Infringing products include Office 2003 and included applications, and Server 2003,			
5		including Microsoft hosted RMS Service using Passport			
6 <sup>.</sup> 7	A method for protecting a first computing	The first computing arrangement surrounded by			
8	arrangement surrounded by a first tamper resistant barrier having a first security level, the method including:	a tamper resistant barrier is the Office 2003 IRM client environment and includes the signed digital certificate identifying the user. If			
9	die medieu meidenig	the certificate is tampered with, or if certain, sensitive IRM processes or modules are debugged or tampered with, the system will			
10		cease to operate.			
11		The first security level is the "Security Level" which has been selected for a particular Office			
12 13	preventing the first computing arrangement	Application, e.g., Word.			
14	from using the same software module accessed by a second computing arrangement having a	The computing arrangement that prevents a software module from running in one			
15	second tamper resistant barrier with a second security level different from the first security	computing arrangement and not in another is the type and characteristics of the particular software module (VBA program within a			
16	level.	document or add-in); i.e., signed, script author, code capabilities, etc., and the "Security Level"			
17		settings.			
18					
19					
20					
21		•			
22					
22 23					
23 24 25					
23 24 25 26					
23 24 25 26 27					
23 24 25 26					

Exhibit B

,		
4	PERSONAL PROPERTY AND INC.	MANAGEMENTAL MENTAL MEN
•	2.	Product Infringing: Windows Media Rights
5		Manager and Windows Media Player
	A system including:	
6	(a) a first apparatus including,	Consumer's computer, as shown in WMRM SDK
.7	(1) user controls,	Consumer's computer, as shown in WMRM SDK
8	(2) a communications port,	Consumer's computer, as shown in WMRM SDK
9	(3) a processor,	Consumer's computer, as shown in WMRM SDK
10	(4) a memory storing:	Consumer's computer, as shown in WMRM SDK
11	(i) a first secure container containing	Secure container (packaged Windows Media
,,	a governed item, the first secure	file), received by consumer's computer from
12	container governed item being at	"Content provider" (WMRM SDK, Step 3), which contains encrypted governed item
13	least in part encrypted; the first secure container having been	("Encrypted content")
	received from a second apparatus;	( Emorypied content )
14	(ii) a first secure container rule at least	Rights portion of signed license, received by
	in part governing an aspect of	consumer's computer from "License issuer"
15	access to or use of said first secure	(WMRM SDK, Step 9)
16	container governed item, the first	
10	secure container rule [sic], the first secure container rule having been	
17	received from a third apparatus	
	different from said second	·
18	apparatus; and	`
	(5) hardware or software used for	Windows Media Player and Windows Media
19	receiving and opening secure	Rights Manager
20	containers, said secure containers each	·
20	including the capacity to contain a governed item, a secure container rule	
21	being associated with each of said	
	secure containers;	
22	(6) a protected processing environment at	1st and 2nd rules consist of any two valid rules
	least in part protecting information	as specified in the Window Media Rights
23	contained in said protected processing	Manager SDK; protected processing
24	environment from tampering by a user	environment includes Windows Media Rights
27	of said first apparatus, said protected	Manager and Windows processes for
25	processing environment including hardware or software used for	protecting operation of Windows Media Rights Manager. Licenses can be used to convey
	applying said first secure container	multiple rules.
26	rule and a second secure container rule	
	in combination to at least in part	·
27	govern at least one aspect of access to	
_	or use of a governed item contained in	
28	a secure container; and	
	(7) hardware or software used for	Any hardware or software employed in

Exhibit B

1

2

transmission of secure containers to other apparatuses or for the receipt of secure containers from other transmitting Windows Media files, including for example consumer's computer's communication port and Windows Media Player (WMRM SDK, Step 3) apparatuses. ·5 

	FOR U.S. PATENT NO. 6,185,683			
3	CLAIM:LANGUAGE	CLAIM OF INFRINGEMENT		
4	2.	Infringing products include Office 2003 and		
5		included applications, and Server 2003, including Microsoft hosted RMS Service using		
6	A greatest including:	Passport		
7	A system including: a first apparatus including,	A device with user controls, a communications		
8	user controls,	port, a processor and memory. For example, the user controls may be a keyboard and		
9	a communications port,	mouse, the communications port may be a NIC card with an Ethernet port, the processor may		
10	a processor,	be a CPU, and the memory may be a hard-drive or RAM.		
11	a memory storing:			
12	a first secure container containing a governed item, the first secure container governed item	An encrypted IRM-governed email received from a remote computer. The encrypted IRM-		
13	being at least in part encrypted; the first secure container having been received from a second	governed email contains an encrypted IRM- governed email message.		
14	apparatus;			
15	a first secure container rule at least in part governing an aspect of access to or use of said	The first secure container rule is received from the RMS server in the form of a use license.		
16	first secure container governed item, the first secure container rule, the first secure container	This use license contains rules generated by the		
17	rule having been received from a third apparatus different from said second	RMS server specifically for the user (or user's group)		
18	apparatus; and hardware or software used for receiving and	The RM-enabled device contains hardware or		
19	opening secure containers,	software for receiving and opening secure emails.		
20	said secure containers each including the capacity to contain a governed item, a secure	The secure email has the capacity to contain an		
21	container rule being associated with each of said secure containers;	IRM-governed email message, with a rule being associated with each email.		
22		The rules associated with the secure emails are		
23		rules that come as part of the original email as well as rules that come back from the RMS.		
24	a protected processing environment at least in part protecting information contained in said	Protected information on the RM-enabled device is protected by the use of at least		
25	protected processing environment from	cryptographic techniques.		
26	tampering by a user of said first apparatus,	The rule governing the email works together		
27	said protected processing environment including hardware or software used for	The rule governing the email works together with an additional rule to determine what		
28	applying said first secure container rule and a second secure container rule in combination to	access to or use (if any) are allowed with respect to the IRM-governed email message.		
	at least in part govern at least one aspect of	For example, the additional rule may be		

Exhibit B

1

secure containers to other apparatuses or for the receipt of secure containers from other apparatuses  for transmitting or receiving secure example, RM-enabled OUTLO to transmit and receive encrypted.		
	The device includes hardware or software us for transmitting or receiving secure emails. Fexample, RM-enabled OUTLOOK is designed to transmit and receive encrypted IRM-governed emails to/from other devices.	
	· -	
	•	
	•	
	•	

293482.02

3		
4	CLAIM/LANGUAGE/4	CLAIM OF INERINGEMENT
5	2.	Infringing products include Office 2003 and included applications, and Server 2003,
6		including Microsoft hosted RMS Service using Passport
7	A system including:	
8	a first apparatus including,	A device with user controls, a communications port, a processor, and memory. For example,
9	user controls,	the user controls may be a keyboard and mouse, the communications port may be a NIC
10	a communications port,	card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive
11	a processor,	or RAM.
12	a memory storing: a first secure container containing a governed	The first secure container is an encrypted IRM-
.13	item, the first secure container governed item being at least in part encrypted; the first secure	protected document.
14	container having been received from a second apparatus;	This encrypted IRM-governed document is, for example, received from a remote computer, as an attachment to an IRM-governed email or
15 16		downloaded from a document server or web site.
17	a first secure container rule at least in part governing an aspect of access to or use of said	The first secure container rule is received from the RMS server in the form of a use license.
.18	first secure container governed item, the first secure container rule, the first secure container	This use license contains rules generated by the
19	rule having been received from a third apparatus different from said second	RMS server specifically for the user (or user's group).
20	apparatus; and	<u> </u>
21	hardware or software used for receiving and opening secure containers,	The RM-enabled device contains hardware or software for receiving and opening secure documents.
22	said secure containers each including the capacity to contain a governed item, a secure	The secure documents have the capacity to
23	container rule being associated with each of said secure containers;	contain IRM-governed content, with a rule being associated with each secure document.
24		The rules associated with said secure
25		documents are the rules that come as part of the originally received document as well as rules
26		that come back from the RMS server.
27	a protected processing environment at least in part protecting information contained in said	Protected information on the RM-enabled device is protected by the use of at least
28	protected processing environment from tampering by a user of said first apparatus,	cryptographic technique.
·		The rule governing the document works
1	l .	1

Exhibit B

293482.02

said protected processing including hardware or so applying said first secure second secure container at least in part govern at access to or use of a govern a secure container; and	oftware used for e container rule and a rule in combination to least one aspect of emed item contained	what access to respect to the I example, the a with an email t	in additional rule or use (if any) a RM-governed d dditional rule m to which the doctored together was a rule of the control o	re allowed wo ocument. For ay be associa ument was
hardware or software use secure containers to othe the receipt of secure con apparatuses.	er apparatuses or for	for transmitting For example, I designed to tra	ludes hardware g or receiving se RM-enabled OU nsmit and receiv with IRM-gove o.	cure docume TLOOK is ve'to/from oth
	• •		•	
			··	
		•		•
• •				•
	•			
		•		
	•		•	
				•
	• •		•	
			•	
		•		
				,
			•	
-				

3	CLADATA STORY	Too begins the begins of the second
1	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5	3.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
7	A system including:	
, 8 9 0	a first apparatus including, user controls, a communications port, a processor,	A device with user controls, a communication port, a processor and memory. For example, the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-driv or RAM.
l	a memory storing:	
) }	a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	The first secure container containing a governed item is an IRM protected email.
). }		Both the email and attachment are IRM protected, each having their own rules, each being encrypted.
	a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	The rule governing the email (a first secure container rule) governs said first secure container governed item.
	a second secure container containing a digital certificate;	The second secure container is the IRM protected attachment's derived license request object.  The license request object contains the Publishing license and a signed digital certificate.
i		
	hardware or software used for receiving and opening secure containers,	The RM (IRM) enabled computer has software for receiving and opening secure containers.
	said secure containers each including the capacity to contain a governed item, a secure	The IRM secure containers have capacity to contain a governed item, with a secure
	container rule being associated with each of said secure containers;	container rule being associated with each of said secure containers.
	a protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus,	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques.
	said protected processing environment including hardware or software used for	The rules governing the email itself (first

Exhibit B

applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	secure container rule) and the rules governing the attachment work together to determine what access to or use (if any) will be allowed with respect to the governed item.
hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	IRM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.
•	·
·	
•	
·	
•	1

293482.02

Exhibit B

3		
4	CLAIMLANGUAGE	A CLAIM OF INFRINGEMENT
5	3.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
7	A system including:	
8	a first apparatus including, user controls,	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and '
9	a communications port,	mouse, the communications port may be a NIC card with an Ethernet port, the processor may
.10.	a processor,	be a CPU, and the memory may be a hard-drive or RAM.
11	a memory storing:	
12	a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	The first secure container containing a governed item is an IRM protected document, which is an attachment within an IRM
13 14	being at least in part enerypted,	protected email message. The governed item is the document's content.
15		Both the email message and attachment are encrypted and have associated usage rules due to IRM protection.
16 17	a first secure container rule at least in part governing an aspect of access to or use of said first secure container governed item; and	A use license for the IRM protected document specifies rules governing access to or use of said first secure container governed item.
18	a second secure container containing a digital certificate;	The second secure container is the IRM protected email message.
19 20		The IRM protected attachment includes a publishing license and an owner certificate,
21		both of which are signed XrML digital certificates.
22 23		The attachment (including embedded certificates) is contained within the IRM protected email message (said second secure
		container).
24	hardware or software used for receiving and opening secure containers,	The RM (IRM) enabled computer has software for receiving and opening secure containers.
<ul><li>25</li><li>26</li></ul>	said secure containers each including the capacity to contain a governed item, a secure	The IRM secure containers have capacity to
27	container rule being associated with each of said secure containers:	contain a governed item, with a secure container rule being associated with each of said secure containers.
28	a protected processing environment at least in part protecting information contained in said protected processing environment from	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques.
.		I I I I I I I I I I I I I I I I I I I

Exhibit B

293482:02

1

2	tampering by a user of said first apparatus,	
3	said protected processing environment including hardware or software used for	The rules governing the attachment (first secure
4	applying said first secure container rule and a second secure container rule in combination to	container rule) and the rules governing the email message (second secure container rule) work together to determine what access to or
5	at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	use (if any) will be allowed with respect to the governed item.
6	hardware or software used for transmission of secure containers to other apparatuses or for	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured
7	the receipt of secure containers from other apparatuses.	containers to/from other computers.
8	4. A system as in claim 3,	
9	said memory storing a rule associated with said second secure container, said rule	All parts of the attachment (including embedded signed XIML licenses/certificates)
0.	associated with said second secure container at least in part governing at least one aspect of	are protected by the enclosing email message and governed by the associated email rules
1	access to or use of said digital certificate.	(second secure container rule).
2		
3		· · · · · · · · · · · · · · · · · · ·
4		
5		
6	·	
7 8		
9		
20		•
21		
22		
23		·
24	··	
25		
26		
27		
28		

	INTERTRUST INFRINGEMENT CHART		
2	FOR U.S. PATE	NT NO. 6,185,683	
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT	
5	5.	Infringing products include Office 2003 and included applications, and Server 2003, including Microsoft hosted RMS Service using Passport	
7	A system including:		
8	a first apparatus including,	A device with user controls, a communications port, a processor and memory. For example, the user controls may be a keyboard and	
	user controls,	mouse, the communications port may be a NIC	
9 10	a communications port,	card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive	
11	a processor,	or RAM.	
**	a memory storing:		
12	a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	first secure container containing a governed item is an IRM protected email.	
13	being at least in part encrypted,	Both the email and attachment are IRM	
14		protected, each having their own rules, each being encrypted.	
15	a first secure container rule at least in part	The rule governing the email (a first secure	
16	governing an aspect of access to or use of said first secure container governed item; and	container rule) governs said first secure container governed item.	
17			
18	a second secure container containing a digital	The second secure container is the IRM	
19	signature, the second secure container being different from said first secure container;	protected attachment's derived license request object.	
		The license request object contains the Publishing license and a signed digital	
20	·	certificate.	
21	hardware or software used for receiving and	The RM (IRM) enabled computer has software	
22	opening secure containers, said secure containers each including the capacity to	for receiving and opening secure containers.	
23	contain a governed item, a secure container rule being associated with each of said secure	The IRM secure containers have capacity to contain a governed item, with a secure	
24	containers;	container rule being associated with each of said secure containers.	
25	a protected processing environment at least in	Protected information on the RM-enabled computer is protected by the use of at least	
26	part protecting information contained in said protected processing environment from tampering by a user of said first apparatus,	cryptographic techniques.	
27	said protected processing environment	·	
28	including hardware or software used for anniving said first secure container rule and a	The rules governing the email itself (first secure container rule) and the rules governing	

Exhibit B

293482.02

2	second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container; and	the attachment will work together to determine what access to or use (if any) will be allowed with respect to the governed item.
3	hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured containers to/from other computers.
5	apparatuses.	
6		
7	•	
8		
9		
10		
11	•	• •
12		•
13		•
14	•	
15	•	:
16		
17	•	•
18		
19		
20		
21		•
22		
23		
24		
25		
26		
26 27		

Exhibit B

293482.02

3		· (
4	CLAIM LANGUAGE	GLAIM OF INFRINGEMENT
5	5.	Infringing products include Office 2003 and included applications, and Server 2003,
6		including Microsoft hosted RMS Service using Passport
7	A system including:	
8	a first apparatus including,	A device with user controls, a communications port, a processor and memory. For example,
9	user controls,	the user controls may be a keyboard and mouse, the communications port may be a NIC
10	a communications port,	card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive
11	a processor,	or RAM.
12	a memory storing:	C
13	a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	first secure container containing a governed item is an IRM protected email.
14	·	Both the email and attachment are IRM protected, each having their own rules, each being encrypted.
15	a first secure container rule at least in part	The rule governing the email (a first secure
16	governing an aspect of access to or use of said first secure container governed item; and	container rule) governs said first secure container governed item.
17		
18	a second secure container containing a digital signature, the second secure container being	The second secure container is the IRM email attachment.
19	different from said first secure container;	This attachment and its publishing license are
20		signed.
21	hardware or software used for receiving and opening secure containers, said secure	The RM (IRM) enabled computer has software for receiving and opening secure containers.
22	containers each including the capacity to contain a governed item, a secure container	The IRM secure containers have capacity to
23	rule being associated with each of said secure containers;	contain a governed item, with a secure container rule being associated with each of
24		said secure containers.
25	a protected processing environment at least in- part protecting information contained in said protected processing environment from	Protected information on the RM-enabled computer is protected by the use of at least cryptographic techniques.
26	tampering by a user of said first apparatus,	11-00-uhun
27	said protected processing environment	m 1 15.00 15
28	including hardware or software used for anniving said first secure container rule and a	The rules governing the email itself (first secure container rule) and the rules governing
ı		#

Exhibit B

293482.02

1

at l	ond secure container rule in combination to east in part govern at least one aspect of ess to or use of a governed item contained secure container; and	the attachment work together to determine what access to or use (if any) will be allowed with respect to the governed item.
har sec the	dware or software used for transmission of ure containers to other apparatuses or for receipt of secure containers from other paratuses.	RM-enabled applications, e.g., OUTLOOK, a designed to transmit and receive RM secured containers to/from other computers.
app	arutoses.	
		•
	·	
ľ.		
		. %
	•	
	·	
	·	
		·
		•
	·	
		· •
	·	·
11	•	•

Exhibit B

293482.02

•	FOR U.S. PATENT NO. 6,185,683	
3		
4		CEAIM OF INFRINGEMENT
5	5.	Infringing products include Office 2003 and
. 6		included applications, and Server 2003, including Microsoft hosted RMS Service using Passport
. 7	A system including:	
8	a first apparatus including,	A device with user controls, a communications port, a processor and memory. For example,
. 9	user controls,	the user controls may be a keyboard and mouse, the communications port may be a NIC
10	a communications port,	card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive
11	a processor,	or RAM.
12	a memory storing:	m c
13	a first secure container containing a governed item, the first secure container governed item being at least in part encrypted;	The first secure container containing a governed item is an IRM protected document, which is an attachment within an IRM
14	par one., proc,	protected email message. The governed item is the document's content.
15		Both the email message and attachment are
16		encrypted and have associated usage rules due to IRM protection.
17	a first secure container rule at least in part governing an aspect of access to or use of said	A use license for the IRM protected document specifies rules governing access to or use of
· 18.	first secure container governed item; and	said first secure container governed item.
19	a second secure container containing a digital signature, the second secure container being different from said first secure container;	The second secure container is the IRM protected email message.
20	,	The IRM protected attachment includes a
21		publishing license and an owner certificate, both of which are signed XrML digital
22		certificates.
23		The attachment (including embedded certificates) is contained within the IRM protected email message (said second secure
24	hardware or coffmen and for	container).
25	hardware or software used for receiving and opening secure containers, said secure containers each including the capacity to	The RM (IRM) enabled computer has software for receiving and opening secure containers.
26	contain a governed item, a secure container	The IRM secure containers have capacity to
27	rule being associated with each of said secure containers;	contain a governed item, with a secure container rule being associated with each of
28	a protected processing environment at least in part protecting information contained in said	Protected information on the RM-enabled- computer is protected by the use of at least
ı		entraction in indicated in the use of at least

Exhibit B

293482.02

3

		1
1		
2	protected processing environment from tampering by a user of said first apparatus,	cryptographic techniques.
3	said protected processing environment including hardware or software used for	The rules governing the attachment (first secure
. 4	applying said first secure container rule and a second secure container rule in combination to	container rule) and the rules governing the email message (second secure container rule)
5	at least in part govern at least one aspect of access to or use of a governed item contained	work together to determine what access to or use (if any) will be allowed with respect to the
6	in a secure container; and hardware or software used for transmission of	RM-enabled applications, e.g., OUTLOOK, are designed to transmit and receive RM secured
7	secure containers to other apparatuses or for the receipt of secure containers from other apparatuses.	containers to/from other computers.
8	6. A system as in claim 5,	
9	said memory storing a rule at least in part governing an aspect of access to or use of said	All parts of the attachment (including embedded signed XrML licenses/certificates)
10	digital signature.	are protected by the enclosing email message and governed by the associated email rules (second secure container rule).
Ì	· · · · · · · · · · · · · · · · · · ·	1 (Second Second Container Faire).
12 13		
14		
15		
16		
17		
18		
19		
20 21	·	
22		
23		
24		
25		
26		
27		
28		
	Exh	nibit B

3		
4	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5	28.	Infringing products include Office 2003 and included applications, and Server 2003,
6		including Microsoft hosted RMS Service using Passport
7	A system including:	
8	a first apparatus including;	A device with user controls, a communications port, a processor and memory. For example,
9	user controls,	the user controls may be a keyboard and mouse, the communications port may be a NIC
10	a communications port,	card with an Ethernet port, the processor may be a CPU, and the memory may be a hard-drive
11	a processor,	or RAM.
12	a memory containing a first rule,	The first rule governs use of an IRM protected document (e.g., an IRM rule permitting a document to be read by specified users or
13		barring access to IRM-governed information from specified users, applications, or other
14		principals).
15	hardware or software used for receiving and opening secure containers,	The RM-enabled device contains hardware or software for receiving and opening secure containers.
16	said secure containers each including the	containers.
17	capacity to contain a governed item, a secure container rule being associated with each of	The secure email has the capacity to contain an IRM-governed email message, with a rule being associated with each email.
18	said secure containers; a protected processing environment at least in	Protected information on the RM-enabled
19	part protecting information contained in said protected processing environment from	device is protected by the use of at least cryptographic techniques.
20	tampering by a user of said first apparatus,	The secure container rule is an IRM rule
21	said protected processing environment including hardware or software used for	governing access to the IRM protected document (e.g., a rule permitting editing by
22	applying said first rule and a secure container rule in combination to at least in part govern at	specified users).
23	least one aspect of access to or use of a	The rule governing the email works together with an additional rule to determine what
24	governed item; and	access to or use (if any) are allowed with
25		respect to the IRM-governed email message (the document's content). For example, the additional rule may be received together with
26		the rule in the use license, may be associated
27		with a publishing license, may be associated with user certification, revocation lists, or
28		exclusion policies, or may be received from any other source.
20	hardware or software used for transmission of	The device includes hardware or software used

Exhibit B 79

293482.02

1

1 2	secure containers to other apparatuses or f r the receipt of secure containers from other apparatuses; and	for transmitting or receiving secure containers. For example, RM-enabled OUTLOOK is designed to transmit and receive encrypted
3		IRM-governed emails to/from other devices.
` ∦	a second apparatus including:	
4	user controls,	A device with user controls, a communications port, a processor and memory. For example,
5	a communications port,	the user controls may be a keyboard and mouse, the communications port may be a NIC card with an Ethernet port, the processor may
6 7	a processor, a memory containing a second rule,	be a CPU, and the memory may be a hard-drive or RAM.
.	a memory containing a second rate,	
8		The second rule governs use of an IRM protected document (e.g., an IRM rule permitting a document to be read by specified
9		users or barring access to IRM-governed
0 ·		information from specified users, applications, or other principals).
1	hardware or software used for receiving and	The RM-enabled device contains hardware or
2	opening secure containers,	software for receiving and opening secure containers.
.	said secure containers each including the	The second amoil has the capacity to contain a
3 4	capacity to contain a governed item, a secure container rule being associated with each of said secure containers;	The secure email has the capacity to contain as IRM-governed email item, with a rule being associated with each secure containers.
* ∦	a protected processing environment at least in	Protected information on the RM-enabled
5	part protecting information contained in said protected processing environment from	device is protected by the use of at least cryptographic technique.
6	tampering by a user of said apparatus,	The secure container rule is an IRM rule
7	said protected processing environment	governing access to the IRM protected
8	including hardware or software used for applying said second rule and a secure	document (e.g., a rule permitting editing by specified users).
9	container rule in combination to at least in part govern at least one aspect of access to or use	The rule governing the email works together
0	of a governed item;	with an additional rule to determine what access to or use (if any) are allowed with
1		respect to the IRM-governed item (the document's content). For example, the additional rule may be received together with
2		the rule in the use license, may be associated
3		with a publishing license, may be associated with user certification, revocation lists, or
4		exclusion policies, or may be received from any other source.
5	hardware or software used for transmission of secure containers to other apparatuses or for	The device includes hardware or software used for transmitting or receiving secure containers.
6	the receipt of secure containers from other apparatuses; and	For example, RM-enabled OUTLOOK is designed to transmit and receive encrypted
7	<del></del>	IRM-governed emails to/from other devices.
• 1)	an electronic intermediary, said intermediary	The RMS Server (Microsoft hosted or
8	including a user rights authority clearinghouse.	otherwise) constructs a 'use license' specific to a piece content and targets it to a specific user.

Exhibit B 80

293482.02

29. A system as in claim 28, said user rights authority clearinghouse operatively connected to make rights available to users.	The RMS server sends use licenses to users through a communications port, e.g., Etherne serial, satellite, "the internet" These use licenses include rights.
	The clearing functionality of the RMS is operatively connected to the RMS server.
•	
	• •
	·
	·
	•
	•
•	

293482.02

Exhibit I

	FOR 0.5. 1 A 1 E 1 1 1 1 0 . 0,105,005		
4	28.	Product Infringing: Windows Media Rights Manager and Windows Media Player	
5	A system including:		
6	(a) a first apparatus including;	Consumer's computer, as shown in WMRM SDK	
7	(1) user controls,	Consumer's computer, as shown in WMRM SDK	
8	(2) a communications port,	Consumer's computer, as shown in WMRM SDK	
9	(3) a processor,	Consumer's computer, as shown in WMRM SDK	
10 ·	(4) a memory containing a first rule,	Memory is in the consumer's computer, first rule is a right received as part of a signed license (WMRM SDK, Step 9)	
12	(5) hardware or software used for receiving and opening secure	Consumer's computer receives Windows Media file (secure container) via	
13	containers, said secure containers each including the capacity to contain		
14	a governed item, a secure container rule being associated with each of said secure containers;	Windows Media Player and Windows Media Rights Manager.	
15 16	(6) a protected processing environment a least in part protecting information contained in said protected processing environment from tampering by a	Media Rights Manager and Windows	
17 18	user of said first apparatus, said protected processing environment including hardware or software used for applying said first rule and a		
19 20	secure container rule in combination to at least in part govern at least one aspect of access to or use of a		
20	governed item; and		
21	(7) hardware or software used for transmission of secure containers to	Hardware or software employed in transmitting Windows Media files, including for example	
22 23	other apparatuses or for the receipt of secure containers from other apparatuses; and	consumer's computer's communication port and Windows Media Player (WMRM SDK, Step 3)	
ا دے	(b) a second apparatus including:	2nd consumer's computer	
24	(1) user controls,	2nd consumer's computer	
[	(2) a communications port,	2nd consumer's computer	
25	(3) a processor,	2nd consumer's computer	
26	(4) a memory containing a second rule,	Memory is in the 2nd consumer's computer, first rule is a Right received as part of a signed license (WMRM SDK, Step 9)	
27	(5) hardware or software used for	2nd consumer's computer receives Windows	
28	receiving and opening secure containers, said secure containers each including the capacity to contain	Media file (secure container) via communications port (WMRM SDK, Step 3) and applies secure container rule or rules via	
H	cach merdding the capacity to contain	i and applies secure container rule of rules via	

Exhibit B

293482,02

I

,				
1 2		a governed item, a secure rule being associated wit said secure containers;	e container h each of	Windows Media Player and Windows Media Rights Manager.
3	(6)	a protected processing er least in part protecting in	formation	Processing environment includes Windows Media Rights Manager and Windows processes for protecting operation of Windows
4 5	·	contained in said protect environment from tampe user of said apparatus; sa	ring by a a did protected	Media Rights Manager; processing environment applies multiple rules in
6		processing environment hardware or software use applying said second rule	ed for	combination
7		secure container rule in c to at least in part govern	combination	
8		aspect of access to or use governed item:	<u> </u>	The second secon
9	(7)	hardware or software use transmission of secure co other apparatuses or for t secure containers from o	ontainers to the receipt of	Hardware or software employed in transmitting Windows Media files, including for example 2 <sup>nd</sup> consumer's computer's communication port and Windows Media Player (WMRM
11		apparatuses; and	·	SDK, Step 3) License Issuer
12	inte	electronic intermediary, sa rmediary including a user nority clearinghouse.		License issuer
13	29. A s	stem as in claim 28,	house	License Issuer, operatively connected to
14	said use operativ	r rights authority clearing ely connected to make rig	hts available	consumer's computer (WMRM SDK, Step 9)
15	to users.			
16				
17		·		
18				
19		•	· •	
20	·			
22				•
23				
24				
25				
26				
27				
28	·			:: ::: :::::::::::::::::::::::::::::::
			Exh	ibit B 33

_	FOR U.S. PATENT NO. 0,105,005	
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
4		
7	56.	Infringing products include Office 2003 and
5		included applications, and Server 2003, including Microsoft hosted RMS Service using
6		Passport
7	A method of securely delivering an item,	
,	including the following steps: performing an authentication step;	The RM-enabled application, e.g., Word,
8	performing all audientication step,	OUTLOOK, PowerPoint, etc., must be
9	·	authenticated before it is allowed access to or
9		use of the content.
10	associating a digital signature with said item;	The RM protected content is signed.  RM-protected content is packaged with rules
	incorporating said item into a first secure electronic container, said item being at least in	and encrypted.
11	part encrypted while in said container,	and cheryprod.
12		
	said incorporation occurring in an apparatus	
13	containing a first protected processing environment, said protected processing	Protected information on the RM enabled
14	environment at least in part protecting	computer is protected by the use of at least cryptographic techniques.
•	information contained in said protected	cryptograpine techniques.
15	processing environment from tampering by a	
16	user of said apparatus; in said protected processing environment,	The IRM-protected document (said item) has
10	associating a first rule with said first secure	an associated rule or rules.
17	electronic container, said first rule at least in	
10	part governing at least one aspect of access to	
18	or use of said item; authenticating an intended recipient of said	A recipient of IRM-protected content must be
19	item;	authenticated before being allowed access to or
	•	use of the content.
20	transmitting said first secure electronic	The document is sent via IRM-protected email as an attachment.
21	container and said first rule to said intended recipient; and	as an attachment.
	using a second protected processing	The email is received at another IRM-enabled
22	environment, providing said intended recipient	computer.
23	access to at least a portion of said item,	
23	said access being governed at least in part by	
24	said first rule and by a second rule present at	The first said rule is the rule(s) associated with
25	said intended recipient's site.	the attached document, and the second rule is
25		the rule(s) received that govern the email itself.
26	· .	

Exhibit B 

_		i
4	126.	Product Infringing: Windows Hardware
5		Quality Labs Authentication services,
		Windows operating Systems (such as Windows XP) that support the driver
6		signing features, and any product using
7	A make d of moviding among intermediant	Driver Signing feature
	A method of providing trusted intermediary services including the following steps:	
8	at a first apparatus, receiving an item from	Microsoft's Window Hardware Quality
9	a second apparatus;	Labs (WHQL) (first apparatus) receiving driver package (item) from independent
		hardware vendor (IHV) or any driver
10		developer (second apparatus).
11	associating authentication information with	The signature information of a security catalog file (see next element of claim)
12	said item;	names Microsoft as the publisher.
12		WHQL's signature is intended to signify
13		that a driver has complied with Microsoft's Windows compatibility and/or Secure
14		Audio Path (SAP) specifications.
17	incorporating said item into a secure digital	The hashes of the files making up the
15	container;	driver package are included in the signed security catalog file for the driver package.
16		The catalog file makes the driver package a
10		secure digital container.
17	associating a first rule with said secure	Driver developers specify rules in an INF file that govern the installation and/or use
18	digital container, said first rule at least in part governing at least one aspect of access	of the driver. For example, as specified in
19	to or use of said item;	the INF, the installation events will vary based on the user's operating system
•		version, which includes architecture, product type and suite. The INF logging
20		rules and can further specify security rules
21		that are evaluated when the driver is used.
22		White Paper - Operating-System
23		Versioning for Drivers under Windows XP
	_	Setup selects the [Models] section to use based on the following rules:
24	. '	_
25	·	If the INF contains [Models] sections for several major or minor operating system
26		version numbers, Setup uses the section
27	·	with the highest version numbers that are not higher than the operating system
21	·	version on which the installation is taking
28		place.

Exhibit B

293482.02

2

1	·	
1 2 · 3	·	If the INF [Models] sections that match the operating system version also include product type decorations, product suite decorations, or both, then Setup selects the section that most closely matches the
4		running operating system.
5		Suppose, for example, Setup is running on Windows XP Professional (which is operating system version 5.1), and it finds
6 7		the following entry in a [Manufacturer] section:
8		%FooCorp%=FooMfg, NT, NT.5, NT.5.5, NT0x80
9		In this case, Setup will look for a [Models]
10 11		section named [FooMfg.NT.5]. Setup will also use the [FooMfg.NT.5] section if it is running on a Datacenter version of
12		Windows .NET Server, because a specific major/minor version takes precedence over the product type and suite mask.
13		For example, to create an INF that is
14		intended for use only on Windows XP, the INF file could contain the following:
15		[Manufacturer]
16		"Foo Corp." = FooMfg, NT.5.1, NT.5.2 [FooMfg.NT.5.1]
17		"Foo Device" = FooDev, *FOO1234
18		Note the omission of the undecorated [FooMfg] section, as well as the omission
19		of the [FooMfg.NT.5.2] section. This INF file would appear to be "empty" on any
20		operating system other than Windows XP.
21		Access Control List Rules
22		XP DDK - Tightening File-Open Security in a Device INF File
23		For Microsoft Windows 2000 and later,
1		Microsoft tightened file-open security in the class installer INFs for certain device
24		classes, including CDROM, DiskDrive, FDC, FloppyDisk, HDC, and
25		SCSIAdapter.
26		If you are unsure whether the class installer for your device has tightened security on
27		file opens, you should tighten security by using the device's INF file to assign a value
28		to the DeviceCharacteristics value name
		in the registry. Do this within an add-

		•
1		registry-section, which is specified using
2	transmitting said secure digital container	the INF AddReg directive.  Microsoft, IHV, driver developer or any
3	and said first rule to a third apparatus, said third apparatus including a protected	other party distributing signed driver packages transmitting the driver package to
4.	processing environment at least in part	user (third apparatus). Since the driver
5	protecting information stored in said protected processing environment from	package includes the INF file, it will include the first rule. The protected
_	tampering by a user of said third apparatus;	processing environment (PPE) is Windows operating system with its pertinent services
6		such as Windows File Protection, signature and cryptographic functions, Plug and Play
7		and Set-up and their related default and
8	•	modifiable policies. The PPE checks for signatures on driver packages and detects
. 9 .		situations when the driver package's signature does not match the driver
10		package.
11		Additionally, the Digital Rights Manager (DRM) components (kernel and client) will
12		contribute to making the third apparatus a PPE when the SAP functionality is
13		invoked. [That is, when SAP is required, an
14		additional signature is checked to verify that the driver is SAP compliant and that it
	said third apparatus receiving said secure	hasn't been tampered with.] The end-user receiving the driver package.
15	digital container and said first rule;	
16	said third apparatus checking said authentication information; and	A step in the Plug and Play/Setup driver installation process checks signature at
17		installation. Additionally, the DRM component will check the DRM signature
18		when invoking DRM functionality.
19		White Paper - Driver Signing for Windows
20	,	During driver installation, Windows
21		compares the hashes contained in the driver's CAT file with the computed hash
22		of the driver binaries to determine whether the binaries have changed since the CAT
23		file was created. If a driver fails the signature check or there is no CAT file,
24		what happens next depends on the driver signing policy in effect on the user's
		system:
25		If the policy is set to Ignore, the driver installs silently, with no message to the
26		user.
27	·	If the policy is set to Warn, a message
28		warns the user the driver is unsigned, which means that it has not passed WHQL
		THE THE THE TABLE THE PERSON WITH

testing and might cause problems. The Warn dialog box gives an administrative 2 user the option to override the warning and install an unsigned driver anyway. 3 If the policy is set to Block, the system displays a message that informs the user that the driver cannot be installed because 5 it is not digitally signed. The action would be installing and/or using said third apparatus performing at least one the driver. For example, installation action on said item, said at least one action policies govern the actions (ignore, warn or being governed, at least in part, by said 7 block) taken based on whether a driver is first rule and by a second rule resident at signed or not and these policies (rule) are said third apparatus prior to said receipt of said secure digital container and said first resident on the third apparatus. Another rule is the "ranking" of available drivers rule, said action governance occurring at when selecting a driver to install. This least in part in said protected processing ranking process includes whether a driver environment. 10 is signed or not. Another rule is the security access rules that the class installer 11 that will be used to install the device has. 12 In the case of DRM, the content will have associated rules governing its use in a SAP-13 complaint environment. These rules (the content license) can be resident at the third 14 apparatus particularly in the case when a user is installing a new (SAP-compliant) 15 device that will render previously acquired content or in the case that acquired content 16 cannot be rendered until the user installs required drivers. 17 For example, when installing: 18 The XP driver ranking process and the 19 modifiable default related to signature state of the driver act as the second rule. 20 The driver will be installed only if the first 21 and second rules validate. 22 Operating-System Versioning for Drivers under Windows XP 23 Default System Policy for Unsigned 24 Drivers 25 If the user installs an unsigned driver for a designated device class from disk or from 26 another web site, Windows XP/Windows 2000 displays a warning that the driver is 27 unsigned, thus helping to preserve the integrity of the released system. However, 28 by default, Windows XP/Windows 2000

does not block installation of unsigned drivers, so vendors can get urgent hot-fixes 2 to customers while waiting for WHQL t 3 test the fix. In Windows XP, the default driver signing policy can be changed through the Hardware tab of the System applet on the Control Panel. A user can change the policy to be more restrictive, but not less restrictive on a per-user basis (that is, a user can change Warn to Block, but not to Ignore). An administrator can change the policy to be either more restrictive or less 8 restrictive for all users on the system by checking "Apply the setting as system default.' 10 Driver Ranking 11 Under Windows XP, the driver ranking strategy has been modified as follows: 12 If an INF file is unsigned, and if neither the 13 [Models] section nor the [DDInstall] section is decorated with an NT-specific 14 extension, the INF file is considered "suspect" and its rank is shifted into a 15 higher range (that is, worse) than all hardware and compatible rank matches of 16 INF files for which one (or both) of those criteria are met. 17 The new ranking ranges will now be: 18 0 - 0xFFF19 (DRIVER HARDWAREID RANK): 20 "trusted" hardware-ID match 0x1000 - 0x3FFF: "trusted" compatible-21 ID match 0x8000 - 0x8FFF: "untrusted" hardware-ID match 22 0x9000 - 0xBFFF: "untrusted" compatible-ID match 23 0xC000 - 0xCFFF: "untrusted" undecorated hardware-ID match (possibly a 24 Windows 9x-only driver) 0xD000 - 0xFFFF: "untrusted" 25 undecorated compatible-ID match (possibly a Windows 9x-only driver) 26 27 127. A method as in claim 126, in which The authentication information will

28

said authentication information at least in part identifies said first apparatus and/or a

The authentication information will identify Microsoft, operator of the first apparatus.

Exhibit B

293482.02

_ :	POR C.S. TA	12111110.0,103,003
3	126.	Products Infringing: Microsoft Software
4.		that includes the Authenticode feature,
7		.NET Framework SDK, Visual Studio,
5		Microsoft technology that supports a digital
_		signature function (such as ActiveX),
6		Windows Installer technology.
	A method of providing trusted intermediary	Infringement is based on use Microsoft ActiveX control, Cabinet file, Microsoft
7	services including the following steps:	Windows Installer, Authenticode and
ا ، ا		Software Restriction Policy technologies.
8		For example, a software publisher
ا و		distributing a signed application that has
"	·	licensed ActiveX controls embedded
10	·	within it would practice this method.
~ i	at a first apparatus, receiving an item from	The item is unsigned software such as an
11	a second apparatus;	ActiveX control or any software packaged
		in a cabinet file or Microsoft Installer (.msi) file. Within the development
12		environment, multiple software developers
,,		(working on a second apparatus) will send
13		their unsigned software to a secure location
14		(first apparatus) containing the entity's
1		private signing key. An example entity
15		would be a software publisher.
		Source: Deploying ActiveX Controls on
16		the Web with the Internet Component
,,		Download
17		
18	·	The holder of the digital certificate
ŀ		Variation variation and fraction of a same
19		Keeping your digital certificate safe is very important. Some firms (including
		Microsoft) do not keep their signature file
20		on site. The signature is kept with the
21		Certificate Authority and files are sent
	· •	there for signing.
22		0: : : : : : : : : : : : : : : : : : :
į	associating authentication information with	Signing the software associates the
23	said item;	software publisher's identify with the software.
24	·	Software.
24	· ,	Source: Packaging ActiveX Controls
25	·	Signing Cabinet Files
	·	A cab file can be digitally signed like an
26		ActiveX control. A digital signature
ľ		provides accountability for software
27		developers: The signature associates a software vendor's name with a given file. A
<u>,,  </u>		signature is applied to a .cab file (or
28	· ,	control) using the Microsoft Authenticode®
ŀ		:

Exhibit B 91

293482.02

1		
		technology. The .cab tool set assists software
2		developers in applying digital signatures to .cab files by allowing a developer to
3		allocate space in the .cab file for the
.4	incorporating said item into a secure digital	signature. Signing software either directly or within a
5	container;	package (cabinet or .msi file) secures it in a digital container.
6	·	Alternately, the signed ActiveX control could be placed into a signed cabinet file.
7	associating a first rule with said secure digital container, said first rule at least in	The first rule would be the licensing support code within the ActiveX control
8	part governing at least one aspect of access	and/or conditional syntax statements when .
9	to or use of said item;	the software is within a signed .msi file. When the software is within a signed
		cabinet file, the first rule can be a rule contained in the software, as is the case
10		when an ActiveX control is packaged in a
11	·	signed cabinet file.
12	·	First rule, in the case of ActiveX:
13		When an application with a licensed ActiveX control is started, an instance of
14		the control usually needs to be created.
	·	The application accomplishes this by making a call to CreateInstanceLic and
15	. •	passing the license key embedded in the application as a parameter in the call. The
16		ActiveX control performs a string comparison between the embedded license
17		key and its own copy of the license key. If
18		the keys match, an instance of the control is created and the application can execute
19		normally.
20		Source: Using ActiveX Controls to
21		Automate Your Web Pages Run-time licensing
·		Most ActiveX Controls should support design-time licensing and run-time
22		licensing. (The exception is the control that
23	. • •	is distributed free of charge.) Design-time licensing ensures that a developer is
24	·	building his or her application or Web page with a legally purchased control; run-time
25		licensing ensures that a user is running an
		application or displaying a Web page that contains a legally purchased control.
26	•	Design-time licensing is verified by control containers such as Visual Basic, Microsoft
27	·	Access, or Microsoft Visual InterDev®.
28	٠.	Before these containers allow a developer to place a control on a form or Web page.
		<u>, ,, ,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,</u>

Exhibit B

1		they first verify that the control is licensed
2		by the developer or content creator. These containers verify that a control is licensed
3	•	by calling certain functions in the control:
	•	If the license is verified, the developer can add it.
4.		Run-time licensing is also an issue for
5		these containers (which are sometimes bundled as part of the final application); the
6		containers again call functions in the control to validate the license that was
7	transmitting said secure digital container	embedded at design time.  The third apparatus is a user computer or
8	and said first rule to a third apparatus, said	an application server. The protected
1	third apparatus including a protected	processing environment (PPE) is Windows operating system, Internet Explorer (IE)
9	processing environment at least in part protecting information stored in said	and pertinent operating IE services such as
10	protected processing environment from	Windows File Protection and security, signature and cryptographic functions
11	tampering by a user of said third apparatus;	related to code signing and related policies.  The PPE checks for signatures on software
12		or the software packages and detects
	·	situations when the signature does not validate as an indication that tampering
13		may have occurred with the item.
14	said third apparatus receiving said secure digital container and said first rule;	Having the third apparatus receiving said secure digital container and said first rule is
15		typical of networked computing environments.
16	said third apparatus checking said	Examine the signature information includes
17	authentication information; and	verifying that signature was creating using the private key that corresponds to the
17		public key of the publisher.
18	said third apparatus performing at least one action on said item, said at least one action	The action would be installation and/or use of the distributed software. The second
19	being governed, at least in part, by said	rule can be software restriction policies
20	first rule and by a second rule resident at said third apparatus prior to said receipt of	resident on the machine, which can be invoked at installation and/or runtime.
21	said secure digital container and said first rule, said action governance occurring at	.NET Framework Security – pg 259
	least in part in said protected processing	
22	environment.	and
23		White Paper – Using Software Restriction Policies in Windows XP and Windows
24		NET Server to Protect Against
25		Unauthorized Software
		Software Restriction Polices is a policy-
26		driven technology that allows administrators to set code-identity-based
27		rules that determine whether an application
28		is allowed to execute. (.NET Framework Security – pg 259)
		1

Exhibit B

	;	
1 2		For example, administrators can set rules for all Windows Installer packages coming from the Internet or Intranet zone.
. 3		As part of the DLL load mechanisms,
4		Software Restriction Policies is invoked and starts to check its most specific rules.
5		Software Restriction Policies get invoked prior to an exe being able to run.
6		The four types of rules are - hash,
. 7		certificate, path, and zone.
8		Note: The hash and certificate rules relate directing to the signature information whereas, the path and zone rules do not.
9		
10	127. A method as in claim 126, in which said authentication information at least in	The software publisher, user of first device, is identified in the authentication
11	part identifies said first apparatus and/or a user of said first apparatus.	information.
12		
13		
14		
15		
16		·
17		
18		
19		
20		
21		
22		
23		
24		
25		· . ·
26		
27	·	
28		
20		

Product infringing: Visual Studio .NET, .NET Framework SDK, Authenticode, .Products that contain the .NET CLR, .Compact CLR or CLI		·	,
A method of providing trusted intermediary services including the following steps: at a first apparatus; receiving an item from a second apparatus;  associating authentication information with said item;  associating a sid item into a secure digital container;  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment at least in part protecting information stored in said protected processing environment at least in part governing by a user of said third apparatus;  asid third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  A method of providing the following steps:  First apparatus is a software build or deployment services computer, or archive fsecond apparatus; is a software build or deployment services computer for antersource, from a developer computer, or archive fsecond apparatus; services computer or anteriors. Protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and	• •	126.	NET Framework SDK, Authenticode,
associating authentication information with said item;  associating a affirst rule with said secure digital container; associating a first rule with said secure digital container, said first rule at least in part protecting information stred in said protected processing environment at least in part protecting information stored in said protected processing environment at least in part protecting information stored in said protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus; said third apparatus receiving said secure digital container and said first rule;  asid third apparatus receiving said secure digital container and said first rule;  asid third apparatus receiving said secure digital container and said first rule;  asid third apparatus checking said authentication information; and  associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a software build or deployment services computer for archive (second apparatus).  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a software build represent the purpose of ensuring the authenticity of the item, along with a messan			
a second apparatus;  a second apparatus;  a second apparatus;  a second apparatus;  associating authentication information with said item;  associating a first rule with said secure digital container;  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  asid third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  ascest to signing key. The item may be a program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft NET members of the Microsoft NET with the bird apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third app	7	services including the following steps:	P'
program, graphic, media object or other resource, from a developer computer, or archive (second apparatus).  Associating authentication information with said item;  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named assembly that contains this assembly and associated attributes.  Including any security demands (such as members of the Microsoft NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  Transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  asaid third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  processing environment from tare producing and extended the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named assembly had associated attributes.  Including any security demands (such as members of the Microsoft NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR distributions.  Associating a current for the purposes.  Producing signed, strongly named associated with the item for the purposes.  Producing signed, strongly name	8		deployment services computer that has
associating authentication information with said item;  Associating a cryptographic hash with the file that will contain this item for the purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Incorporating said item into a secure digital container;  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  code AccessSecurity Attribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR, CLI and/or compact CLR distributions.  Associating a cryptographic bash with the file that will contain this item for the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Producing signed, strongly named assembly and associated attributes.  Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR, CLI distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  Said third apparatus checking said authentication information; and  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process inclu	. 9		program, graphic, media object or other resource, from a developer computer, or
purpose of ensuring the authenticity of the item, along with names and attributes that are desired to be associated with the item for identification purposes.  Incorporating said item into a secure digital container;  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  asid third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  purpose of ensuring the authenticity of the item, along with names and attributes that are included associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR.  Information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying tha	10		Associating a cryptographic hash with the
are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  asid third apparatus receiving said secure digital container and said first rule;  asid third apparatus receiving said secure digital container and said first rule;  asid third apparatus receiving said secure digital container and said first rule;  are desired to be associated with the item for identification purposes.  Producing signed, strongly named associated attributes.  Including any security demands (such as members of the Microsoft NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR.  Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using		said item;	purpose of ensuring the authenticity of the
incorporating said item into a secure digital container;  associating a first rule with said secure digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  Itransmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  asid third apparatus receiving said secure digital container and said first rule;  Producing signed, strongly named assembly that contains this assembly and associated attributes.  Including any security demands (such as members of the Microsoft .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly.  The third apparatus is a user computer or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI' and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC). verifies the strong name of assembles. This process includes verifying that signature was creating using			are desired to be associated with the item
digital container, said first rule at least in part governing at least one aspect of access to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus; protected from tampering broad user of said third apparatus; protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information; and  digital container, said first rule at least in part protected processing environment is windows NT and the .NET CLR, CLI and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework SDK Public Class CodeAccessSecurityAttribute) as part of the assembly an user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application server. The third apparatus is a user computer or an application se			assembly that contains this assembly and
to or use of said item;  transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  computer of an application server. The third apparatus's protected processing environment from tampering by a user of said third apparatus;  code on server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said authentication information: and  said third apparatus checking said authentication information: and  transmitting said secure digital container or an application server. The third apparatus's protected processing environment is Windows NT and the .NET CLR, CLI and/or compact CLR. Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected from tampering because user is not administrator.  The third apparatus is a user	15		members of the Microsoft .NET
transmitting said secure digital container and said first rule to a third apparatus, said third apparatus including a protected processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  20 21 22 23 24 25 26 27 28 28 28 29 20 20 20 21 22 23 24 25 26 27 28 28 28 28 29 20 20 20 20 21 22 23 23 24 25 26 27 28 28 28 28 28 29 20 20 20 21 22 23 23 24 25 26 26 27 28 28 28 28 28 28 28 28 28 28 28 28 28			CodeAccessSecurityAttribute) as part of
processing environment at least in part protecting information stored in said protected processing environment from tampering by a user of said third apparatus;  21	•	and said first rule to a third apparatus, said	The third apparatus is a user computer or an application server. The third
protected processing environment from tampering by a user of said third apparatus;  Information is protected from tampering because user is not administrator, user runs code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI' and/or compact CLR distributions.  Having the third apparatus receiving said secure digital container and said first rule; secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	19	processing environment at least in part	environment is Windows NT and the .NET
code on server, a share on another computer, or over a network. Further this information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Said third apparatus receiving said secure digital container and said first rule;  said third apparatus checking said secure digital container and said first rule is typical of networked computing environments.  Said third apparatus checking said authentication information: and  The .NET Framework, when the assembly is installed into the global assembly cache (GAC). verifies the strong name of assemblies. This process includes verifying that signature was creating using	20	protected processing environment from	Information is protected from tampering
information is protected by a number of protection mechanisms that are included with the Windows NT and CLR, CLI and/or compact CLR distributions.  Said third apparatus receiving said secure digital container and said first rule;  Said third apparatus checking said secure digital container and said first rule is typical of networked computing environments.  The .NET Framework, when the assembly is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	21	tamporing of a coor or one ame approve,	code on server, a share on another
24 said third apparatus receiving said secure digital container and said first rule; 25 said third apparatus checking said secure digital container and said first rule is typical of networked computing environments. 26 said third apparatus checking said authentication information; and 27 (GAC). verifies the strong name of assemblies. This process includes verifying that signature was creating using	22		information is protected by a number of protection mechanisms that are included
digital container and said first rule; secure digital container and said first rule is typical of networked computing environments.  Said third apparatus checking said authentication information: and is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	23		and/or compact CLR distributions.
typical of networked computing environments.  Said third apparatus checking said authentication information: and is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	24		secure digital container and said first rule is
said third apparatus checking said authentication information; and is installed into the global assembly cache (GAC), verifies the strong name of assemblies. This process includes verifying that signature was creating using	25		typical of networked computing environments.
assemblies. This process includes verifying that signature was creating using	26	said third apparatus checking said authentication information; and	The .NET Framework, when the assembly is installed into the global assembly cache
	27		assemblies. This process includes
n	28		

Exhibit B

293482.02

1

1		public key of the publisher.
2	said third apparatus performing at least one action on said item, said at least one action	The action is executing code that is the item or using code that renders the item.
. 3	being governed, at least in part, by said	Action is governed by security demands on
4	first rule and by a second rule resident at said third apparatus prior to said receipt of	code that calls the item or on code that calls code included in the .NET assembly that
•	said secure digital container and said first	manages said item. The second rule is the machine, enterprise, user, and application
5	rule, said action governance occurring at least in part in said protected processing	configuration file resident rules. Typically
6	environment.	these configuration files will be populated before the arrival of most new assemblies
7	·	in a virtual distribution environment. This
8	·	action governance occurs in the protected processing environment of the CLR, CLI
		and/or compact CLR.
9	127. A method as in claim 126, in which	The authentication information will identify the .NET Assembly Class
10	said authentication information at least in part identifies said first apparatus and/or a	company name and trademark attributes
11	user of said first apparatus.	that identify the apparatus or user of the first apparatus as being a member of an
12		entity or a branded source (brand name).
13		
14		
15		
16	·	
17.		
18		
19		
20		
21		
	·	
22		
23		
24		
25		
26		•
27		
21		

3		
· 4	126.	Product infringing: Visual Studio .NET,
5		NET Framework SDK, Authenticode, Products that contain the NET CLR,
6	A method of providing trusted intermediary	Compact CLR or CLI.
7	services including the following steps:	The investment NET occurrence
8	at a first apparatus, receiving an item from a second apparatus;	The item is an unsigned .NET assembly, which can include, but not be limited to, a Web control, multi-file assembly or
9		component. Within the development environment, multiple assembly builders
10		(working on a second apparatus) will send their unsigned assembly to a secure
11		location (first apparatus) containing the entity's private signing key. An example
12		entity would be a software publisher.
13	,	NET Security Framework - pg 130-1
14		Describes this exact practice and further explains the "Delay Signing Assemblies" feature of .NET that accommodates the fact
15 <sub>16</sub>		that "many publishers will keep the private key in a secure location, possibly
17		embedded in specially designed cryptographic hardware."
18		"Delay signing is a technique used by developers whereby the public key is added
19		to the assembly name as before, granting the assembly its unique identity, but no
20		signature is computed. Thus, no private key access is necessary."
21	associating authentication information with said item;	Strong naming the assembly binds the entity's/publisher's name into the
22	, <del></del>	assembly. The public portion of the key used to strongly name the assembly is
23		placed in the assembly manifest. Other assemblies or applications can contain
24		references to the strong names of strongly named assemblies such as in the case of
25		applications that contain references to a set of compliant .NET core libraries. Strong
26	·	naming compliant .NET core libraries with the European Computers Manufactures
27		Association's (ECMA) key is a way to
28		allow any publisher to develop compliant .NET core libraries that can be
-0		authenticated by other applications,

Exhibit B

293482.02

1

1		
1		104
2		NET Security Framework - pg 124 "Strong naming is a process whereby an
		assembly name can be further qualified by
· 3	·	the identity of the publisher."
4		NET Security Framework - pg 133
		The publisher must advertise its public key or keys in an out-of-band fashion (such as
5	·	documentation shipped with the product or
6		on the company Web site)
		NET Security Framework - pg 130 The goal of the ECMA key is to allow a
7		slightly more generalized strong name
8		binding than usual, namely allowing
	· ·	binding to the publisher of the runtime in
9	incorporating said item into a secure digital	use, rather than to a fixed publisher.  Signing the assembly places it in a secure
10	container;	container.
10		NET Framework Security - pg 527
11		Strong named assemblies cannot be modified in any manner without destroying
12		the strong name signature.
12		Applied Microsoft NET Framework
13		Programming - pg 89 Strongly Named Assemblies Are Tamper-
14		Resistant
,,		When the assembly is installed into the GAC, the system hashes the contents of the
15		file containing the manifest and compares
16		the hash value with the RSA digital
		signature value embedded within the PE file (after unsigning it with the public key).
17		If the values are identical, the file's
. 18		contents haven't been tampered with and
		you know that you have the public key that corresponds to the publisher's private key.
19	,	In addition, the system hashes the contents
20	•	of the assembly's other files and compares
		the hash values with the hash values stored in the manifest file's FileDef table. If any
21		of the hash values don't match, at least one
22		of the assembly's files has been tampered
		with and the assembly will fail to install into the GAC.
23	associating a first rule with said secure	A .NET assembly includes imperative and
24	digital container, said first rule at least in	declarative statements/rules that will
	part governing at least one aspect of access to or use of said item;	govern its access or use. For example, role-based security or strong name
25	to or use or said item;	demands in the assembly can be the first
26		rule.
	-	MSDN on Role-Based Security
27		1
28		Applications that implement role-based security grant rights based on the role
		Security Frant Henry 08250 On the fole
••	1	•

	i ·	
. 1		associated with a principal object. The
2		principal object represents the security
<u> </u>		context under which code is running. The
3		PrincipalPermission object represents the identity and role that a particular principal.
		class must have to run. To implement the
4		Principal Permission class imperatively,
5		create a new instance of the class and
•		initialize it with the name and role that you
. 6		want users to have to access your code.
. 7	·	MSDN on StrongNameIdentityPermission
		Stranghlama Idantity Parmission aloss
8		StrongNameIdentityPermission class defines the identity permission for strong
9	•	names. StrongNameIdentityPermission
7		uses this class to confirm that calling code
10		is in a particular strong-named assembly.
	transmitting said secure digital container	The third apparatus is a user computer or
11	and said first rule to a third apparatus, said	an application server. The software
12	third apparatus including a protected	publisher transmitting the .NET assembly
12	processing environment at least in part	to an end-user with a CLR. The third
13	protecting information stored in said	apparatus's protected processing environment is Windows NT and the .NET
	protected processing environment from tampering by a user of said third apparatus;	CLR, CLI and/or compact CLR.
14	tampering by a user of said and appearies,	Information is protected from tampering
15	·	because user is not administrator, user runs
	·	code on server, a share on another
16	·	computer, or over a network. Further this information is protected by a number of
17		protection mechanisms that are included
17		with the Windows NT and CLR, CLI
18	·	and/or compact CLR distributions.
	said third apparatus receiving said secure digital container and said first rule;	The end-user receiving the signed assembly.
19	said third apparatus checking said	The NET Framework, when the assembly
20	authentication information; and	is installed into the global assembly cash
		(GAC), verifies the strong name of
21		assemblies. This process includes
22		verifying that signature was creating using the private key that corresponds to the
22		public key of the publisher.
23		Applied Microsoft .NET Framework
		Programming - pg 89
24		Strongly Named Assemblies Are Tamper- Resistant
25	·	As above.
23		,
26		.NET Framework Security - pg 128
		The verification of any strong name
27		The verification of any strong name assemblies is performed automatically
28		when needed by the .NET Framework.
20		Any assembly claiming a strong name but

1		failing verification will fail to install into the global assembly or download cache or will fail to load at runtime.
3	said third apparatus performing at least one action on said item, said at least one action	Within the CLR (protected processing
4	being governed, at least in part, by said	environment), the execution of the program will depend upon whether the user is of the
5	first rule and by a second rule resident at said third apparatus prior to said receipt of	"role" required of the assembly or whether the calling assembly is from a strong-
6	said secure digital container and said first rule, said action governance occurring at	named assembly specified in the "item" assembly (alternate first rules) and only if
_	least in part in said protected processing environment.	assembly complies with the local code
. 7		access security policy (second rule), as an example of one of the types of rules that
8		.NET Framework allows to be resident on the third apparatus
9.	127. A method as in claim 126, in which	The user of the first apparatus is the developer
10	said authentication information at least in part identifies said first apparatus and/or a	at the assembly developer. Strong naming binds the publisher's name to assembly.
11	user of said first apparatus.	omes the publisher shall to assembly.
12	LaMacchia, Brian, etc, .NET Framework Sec	curity, Addison-Wesley, 2002
13	Richier, Jenrey, Applied Microsoft NET Fra	amework Programming, Microsoft Press, 2002
14		
15		
16		
17		
18		
19		·
20		· · · · · · · · · · · · · · · · · · ·
21		
22		
23		
24	:	·
25	·	
26		
İ		•
27		
28	·	

3		ENT NO. 6,253,193
٠	*********CLAIMILANGUAGE***	ESTRECE MINIOPINERINGEMENTES
. 5		Infringing products include Windows Media Player and Windows Media Rights Manager SDK
	A method comprising:	
· 7	(a) receiving a digital file including music;	Reference is made to the Windows Media Rights Manager SDK Programming Reference ("WMRM SDK"), attached hereto as Exhibit
8		A. Media Player infringement analysis is set forth herein using the example of a music file downloaded and transferred to a portable audio
9 10		player. Consumer receives a Windows Media file (WMRM SDK, Step 3)
11	(b) storing said digital file in a first secure memory of a first device;	Windows Media file is stored in consumer's computer and all use of it is securely managed by the Secure Content Manager in Windows
12	(c) storing information and in its	Media Player.
13	(c) storing information associated with said digital file in a secure database stored on said	License is stored in the License Store (WMRM
•	first device, said information including at least	SDK, Step 5); license includes Rights which may include AllowTransfertoNonSDMI,
14	one budget control and at least one copy	AllowTransfertoSDMI, (or Allow Transfer to
15	control, said at least one budget control including a budget specifying the number of copies which can be made of said digital file;	WM-D-DRM-Compliant devices or other types of devices), and TransferCount- the
16	and said at least one copy control controlling the copies made of said digital file;	number of times a piece of content may be transferred to the device (a transfer budget).
17	(d) determining whether said digital file may be copied and stored on a second device based on at least said copy control;	Windows Media Rights Manager enforces the license restrictions
19	(e) if said copy control allows at least a portion of said digital file to be copied and stored on a	Windows Media Rights Manager determines whether the AllowTransferToNonSDMI or
20	second device,	AllowTransferToSDMI rights are present.(Or, Allow Transfer to WM-D-DRM-Compliant devices or other types of devices.)
21	(1)copying at least a portion of said digital	Transfer to the SDMI or non-SDMI portable
22	file;	device (Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), if allowed by Windows Media Rights Manager
23	(2)transferring at least a portion of said	Portable device necessarily includes at least a
24	digital file to a second device including a memory and an audio and/or video output;	memory and audio output
25	(3)storing said digital file in said memory of said second device; and	Music file is transferred to the portable device
26	(4)including playing said music through said audio output.	Portable device plays the music
27	2. A method as in claim 1, further comprising:	
28	(a) at a time substantially contemporaneous with said transferring step, recording in said	Counter reflecting TransferCount is decremented by Windows Media Rights
	Exhi	bit B

...Exhibit B

293482.02

1

first device informat transfer has occurred	ion indicating that said	Manager
3. A method as in c	laim 2, in which:	
(a) said information transfer has occurred	indicating that said includes an encumbrance	Counter decrement reduces the allowable number of budgeted transfers
on said budget.  4. A method as in c	laim 3. in which:	
(a) said encumbrance number of copies of	e operates to reduce the said digital file authorized	Counter decrement reduces the allowable number of budgeted transfers
by said budget.	·	
	•	
	·	
	•	
		*
		,
·		
		•
		<u>.</u>
	•	
,		
	•	

293482.02

Exhibit B 102

3	INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,253,193		
4 5		Infringing products include Windows Media Player and Windows Media Rights Manager SDK	
6	11. A method comprising:		
	(a) receiving a digital file;	Consumer receives a Windows Media file (WMRM SDK, Step 3)	
7	(b) storing said digital file in a first secure	Windows Media file is stored in consumer's computer and all use of it is securely managed	
8	memory of a first device;	by the Secure Content Manager in Windows Media Player.	
9	(c) storing information associated with said	License information is stored in the License	
10	digital file in a secure database stored on said first device, said information including a first	Store (WMRM SDK, Step 10), license information includes Rights. License Rights	
11	control;	may include AllowTransferToNonSDMI, AllowTransferToSDMI (Allow Transfer to	
12	·	WM-D-DRM-Compliant devices or other types of devices), TransferCount	
13	(d) determining whether said digital file may be copied and stored on a second device based	WMRM determines whether transfer rights are included in license (WMRM SDK, Step 5)	
14	on said first control,	Portable Device Service Provider Module	
15	(1) said determining step including identifying said second device and determining whether said first control	identifies the portable device as either SDMI- compliant or non-SDMI-compliant (or WM-D-	
16	allows transfer of said copied file to said second device, said determination	DRM Compliant or other types of supported devices) and provides this information to	
17	based at least in part on the features	Windows Media Device Manager, which allows the transfer based on whether the device	
18	present at the device to which said copied file is to be transferred;	identification matches the License Right.	
10	(e) if said first control allows at least a portion	If Windows Media Rights Manager determines	
19	of said digital file to be copied and stored on a	whether the AllowTransferToNonSDMI or	
	second device,	AllowTransferToSDMI rights are present (or	
20		Allow Transfer to WM-D-DRM-Compliant devices or other types of devices), the	
21		following steps are performed:	
21	(1) copying at least a portion of said	Transfer to the SDMI or non-SDMI (Allow	
22	digital file;	Transfer to WM-D-DRM-Compliant or other) portable device, if allowed by Windows Media	
23	·	Rights Manager	
24	(2) transferring at least a portion of said digital file to a second device	Portable device necessarily includes at least a memory and audio output	
25	including a memory and an audio and/or video output;		
•	(3) storing said digital file in said memory	Music file is stored in the portable device	
26	of said second device; and	Portable device plays the music	
27	(4) rendering said digital file through said output.	Fortable device plays the music	
28			

Exhibit B

1

-	<u>INTERTRUST TECHNOLOGIE</u>	S CORP. v. MICROSOFT CORP.
3	INTERTRUST INFRINGEMENT CHART	
١	FOR U.S. PATENT NO. 6,253,193	
5		Product infringing: Windows Media Player, Windows Media Player, Windows Media
٠		Rights Manager SDK
6	15. A method comprising:	
7	(a) receiving a digital file;	Consumer receives a Windows Media file ((WMRM SDK, Step 3)
' [	(b) an authentication step comprising:	
8	(1) accessing at least one identifier associated with a first device or with a	License includes identity of user's Windows Media Player. WM Players capable of playing protected content must be individualized.
9	user of said first device; and	They contain a unique (Individualized) DRM client component to which protected WMA
11		content licenses are bound. Content licenses are bound to this DRM individualization
12		module as the result of a challenge sent from the Client to the WMLM service. The
13		challenge contains information about Individualized DRM Client (in the form of an encrypted Client ID) and capabilities of the
14		machine (e.g. support for Secure Audio Path (SAP), version of the WMRM SDK supported
,,		in the player.
15	(2) determining whether said identifier is	Music file cannot be used unless identifier
16	associated with a device and/or user authorized to store said digital file;	indicated in License matches user's Windows Media Player identifier (that is, the Individualized DRM Client to which the
17		license is bound must be the same one supported by the device).
18 19	(c) storing said digital file in a first secure memory of said first device, but only if said	Music file will not be processed through Windows Media Player, including protected
20	device and/or user is so authorized, but not proceeding with said storing if said device	rendering buffers, unless the identifiers match. Protected WMA file can be stored on client
21	and/or user is not authorized;	even if unauthorized but it cannot be decrypted and enter into the secure boundary (first secure
22		memory) of the player unless appropriately licensed.
23	(d) storing information associated with said digital file in a secure database stored on said	License includes Rights and is stored in the License Store, Rights may include AllowTransferToNonSDMI,
24	first device, said information including at least one control;	Allow Transfer To SDMI, (or Allow Transfer To WM-D-DRM-Compliant Device or other
25		device) TransferCount
26	(e) determining whether said digital file may be copied and stored on a second device based	Windows Media Rights Manager enforces the license restrictions
27	on said at least one control:  (f) if said at least one control allows at least a	If appropriate rights are present, the following
28	portion of said digital file to be copied and stored on a second device.	steps are performed:
٠.	(1) copying at least a portion of said	Transfer to the SDMI or non-SDMI (or WM-
		ibit B

293482.02

digital file;		D-DRM Compliant or other) portable device, i allowed by Windows Media Rights Manager
digital file	g at least a portion of said to a second device memory and an audio	Portable device necessarily includes at least a memory and audio output
and/or vide	eo output;	Music file is stored in the portable device
of said sec	d digital file in said memory ond device; and	
(4) rendering output.	said digital file through said	Portable device plays the music
	claim 15, in which:	Protected Windows Media File is encrypted.
form;	to the state of th	WMP will not decrypt file until license is
and further compris	sing:	processed. Licenses are bound to Individualization DLLs, which are bound to
decrypting said dig		Hardware ID. Ind. DLL and Hardware ID must be verified as the Ids to which the license
authentication step	and before said step of file in said memory of said	is bound – this is the authentication process. (Recall that this module was created based in
first device.		part on receipt of the Client Hardware ID or fingerprint and the license was create based in
		part on receipt of a challenge from the client indicating the security properties (SAP-ready,
·		SDK support, etc.) of the client).
·		
	·	•
	e .	
	·	
	•	
		·
I S		
		•

_	FOR 0.5.1A1EN1 NO. 0,255,175	
3	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
4 .	19.	Infringing products include Office 2003 and included applications, and Server 2003,
5		including Microsoft hosted RMS Service using
6	A	Passport
7	A method comprising: receiving a digital file at a first device;	Receiving a digital file such as a Word
8		Document, email, Excel spreadsheet, PowerPoint presentation, or other content at a
9		recipient's device. Such content may be received via email, received on removable
10		media, such as floppy disk, downloaded and viewable by Internet Explorer, e.g., a web page
11		possibly containing graphics and/or audio data, etc.
12	establishing communication between said first device and a clearinghouse located at a	If the digital file is subject to rights management, and the recipient tries to open the
13	location remote from said first device;	digital file in an IRM-enabled application, the IRM-enabled application contacts a remote RMS, i.e., clearinghouse for a use license.
14	said first device obtaining authorization	If the recipient is authorized to access or use
15	information including a key from said clearinghouse;	the digital file, the RMS creates a license for the digital file. The RMS then seals a key
16		inside the license so that only the recipient canaccess or use the digital file. Finally, the RMS sends the license back to the recipient.
17	said first device using said authorization	The recipient's device then uses the key in the license to gain access or decrypt a portion of
18	information to gain access to or make at least one use of said first digital file, including	the digital file.
19	using said key to decrypt at least a portion of said first digital file; and	
20	receiving a first control from said clearinghouse at said first device;	The license received from the RMS at the recipient's device contains at least one control,
21		such as restricting the ability to print, forward, or edit.
22	storing said first digital file in a memory of said first device;	The digital file is stored in the memory of the said recipient's device, such as in RAM, on a
23		hard drive, etc.  The at least one control in the license limits
24	using said first control to determine whether said first digital file may be copied and stored on a second device;	copying the digital file.
25	,	Such controls are set when the digital file was authored. For example, when the digital file is
26		authored, the IRM-enabled application presented the author with a list of policy
27		templates with different rights levels. The
28		author selected an appropriate rights level which may for instance, allow other users in the system to onen and read the document, but not
		The state of the s
	.1	<b>1</b> 1

Exhibit B

293482.02

•		·
2		to modify it, copy text from it, or forward it. These rights or controls are then associated with the digital file.
3		When an attempt is made to access the digital
<b>4 5</b>		file, the RMS determines the recipient's rights based on the recipient's identity and the policies or controls associated with the digital
6	·	file.
7	if said first control allows at least a portion of said first digital file to be copied and stored on a second device.	If the control in the license allows copying the digital file to a second device, then at least a
8	copying at least a portion of said first digital	portion of the digital file is copied, such as by transferring or forwarding the digital
9	file; transferring at least a portion of said first	file in an email message;  A portion of the digital file is then transferred
10	digital file to a second device including a memory and an audio and/or video output:	to a second device, such as a personal computer or portable device. The second device includes
11		a memory and an audio and/or video output.
12		The memory may be a hard-drive, RAM, CD, DVD, or other storage. The audio and/or video
13	storing said first digital file portion in said	output may be speakers and/or a video monitor.  The digital file is stored in the second device's
14	memory of said second device; and rendering said first digital file portion through	memory.  The digital file is rendered through the output,
15	said output.	such as played through the speakers and/or displayed on the video monitor. For example, a
15	i ·	
16		Word document is displayed on the screen of
16 17 18		Word document is displayed on the screen of
17		Word document is displayed on the screen of
17 18		Word document is displayed on the screen of
17 18 19		Word document is displayed on the screen of
17 18 19 20		Word document is displayed on the screen of
17 18 19 20 21		Word document is displayed on the screen of
17 18 19 20 21 22		Word document is displayed on the screen of
17 18 19 20 21 22 23		Word document is displayed on the screen of
17 18 19 20 21 22 23 24		Word document is displayed on the screen of
17 18 19 20 21 22 23 24 25		Word document is displayed on the screen of
17 18 19 20 21 22 23 24 25 26		Word document is displayed on the screen of

- 293482.02

3	INTERTRUST INFRINGEMENT CHART	
ر	FOR U.S. PATENT NO. 6,253,193	
4		Infringing products include Windows Media Player, Windows Media Rights Manager SDK
5	19. A method comprising:	Trayer, Windows Media Togras Manager 5225
_	(a) receiving a digital file at a first device;	WMRM SDK, Step 3.
6	(b) establishing communication between said	WMRM SDK, Step 6.
_	first device and a clearinghouse located at	
7	a location remote from said first device;	
8	(c) said first device obtaining authorization	WMRM SDK, Step 9. [License contains the
0	information including a key from said	key]
9	clearinghouse;	
	(d) said first device using said authorization	WMRM SDK, Step 11.
10	information to gain access to or make at	
	least one use of said first digital file,	
11	including using said key to decrypt at least	·
	a portion of said first digital file; and	
12	(e) receiving a first control from said	WMRM SDK, Steps 8-9.
	clearinghouse at said first device;	
13	(f) storing said first digital file in a memory of said first device;	WMRM SDK, Step 3.
14	(g) using said first control to determine	At least the following WMRMRights Object
	whether said first digital file may be	properties meet this limitation:
15	copied and stored on a second device;	AllowTransferToNonSDMI,
		AllowTransferToSDMI (or AllowTransfer T
16		WM-D-DRM-Compliant Device or other) and
	0) (6 116 4 4 1 11 4 1 4 1 4 1	TransferCount This and all subsequent claim steps occur when
17	(h) if said first control allows at least a portion	the condition specified in the WMRMRights
18	of said first digital file to be copied and	Object property is met
10	stored on a second device.  (i) copying at least a portion of said first	Transfer to the SDMI or non-SDMI (or WM-
19	(i) copying at least a portion of said first digital file;	D-DRM Compliant) portable device, if
17	oigiai inc,	allowed by Windows Media Rights Manager
20	(i) transferring at least a portion of said first	Portable device necessarily includes at least a
~~	digital file to a second device including a	memory and audio output
21	memory and an audio and/or video output;	
	(k) storing said first digital file portion in said	Music file is stored in the portable device
22	memory of said second device; and	•
	(l) rendering said first digital file portion	Portable device plays the music
23	through said output.	
24		

•		
3		· ,
4		Infringing products include Windows Media Player, Windows Media Player, Windows Media Rights Manager SDK
5	51 A method comprising:	1,100.12 / 0,101.101
6	51. A method comprising:  (a) receiving a digital file at a first device;	WMRM SDK, Step 3.
7	(b) establishing communication between said first device and a	WMRM SDK, Step 6.
8	clearinghouse located at a location remote from said first device;	
9	(c) said first device obtaining authorization information from said	WMRM SDK, Step 9,
10	clearinghouse; and (d) said first device using said	WMRM SDK, Step 11.
11	authorization information to gain access to or make at least one use of said first digital file;	
12	(e) storing said first digital file in a memory of said first device;	WMA file stored on client
13	(f) using at least a first control to	If device is based on WM D-DRM, it has a
14	determine whether said first digital file may be copied and stored on a second	certificate that is used to identify the device as compliant as well as the device's security level. The security level indicates support on
15 16	device, said determination based at least in part on (1) identification information regarding said second device, and (2)	the device for such attributes as an internal clock.
17	the functional attributes of said second device;	
18	(g) if, based at least in part on said identification information, said first control allows at least a portion of said	If License specifies that transfer of protected WMA file to WM-D-DRM-Compliant device is allowed, transfer may occur.
19	first digital file to be copied and stored on a second device,	is another, a disserting
20 21	(h) copying at least a portion of said first digital file;	If transfer is a licensed right as indicated in the license, the song is copied to the device via Windows Media Device Manager.
	(i) transferring at least a portion of said	Windows Media Device Manager transfers the content to the device:
22	first digital file to a second device including a memory and an audio	content to the device.
23	and/or video output;	WMA file is stored on device
24	(j) storing said first digital file portion in said memory of said second device; and	WIVEY THE IS Stoled on device
25	(k) rendering said first digital file	WMA file is rendered.

2627

2

28

. 😲

.3		
	MANGUAGE 222	WASHE WICEAIM OF INFRINGEMENT
4	33.	Infringing products include all Microsoft tools that support the Microsoft ActiveX
5		licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating
6		System products that include the Microsoft Installer technology.
7	A data processing arrangement comprising at least one storing arrangement that at	The first protected data is an ActiveX control.
8	least temporarily stores a first secure container comprising first protected data	The first alternative for the first secure
9	and a first set of rules governing use of said first protected data,	container is the signed .msi in which the ActiveX developer packaged the ActiveX
10 11		control. The first set of rules is the conditional syntax statements of the signed .msi file.
12		The second alternative for the first secure
13		container is the signed and licensed ActiveX control. The first set of rules is the license support code in the ActiveX
14		control.
15		A third alternative for the first container is a signed cabinet file containing a (signed or
16 17	·	unsigned) ActiveX control with license support code. The first set of rules is the license support code in the ActiveX
18		control.
19	and at least temporarily stores a second	The second protected data is the application
20	secure container comprising second protected data different from said first	developer's application that includes/uses the ActiveX control. The application
21	protected data and a second set of rules governing use of said second protected	developer's signed .msi file (second secure container) contains the application (second
22	data; and	protected data). The second set of rules is the signed .msi file's conditional syntax
23		statements that will be governed the offer/installation of the application.
24	a data transfer arrangement, coupled to at least one storing arrangement, for	Placing the licensed ActiveX control (first protected information) in a signed cabinet
25	transferring at least a portion of said first protected data and a third set of rules	file (third secure container) that itself is included in the application's signed .msi
26	governing use of said portion of said first protected data to said second secure	file (second secure container). The third set of rules is the license support code in
	container,	the ActiveX control.
27	further comprising	
28	means for creating and storing, in said at least one storing arrangement, a third secure container;	The ability of the application developer to package files in signed cabinet files.
		1-

Exhibit B

. 293482.02

1 2 3 4	said data transfer arrangement further comprising means for transferring said portion of said first protected data and said third set of rules to said third secure container, and means for incorporating said third secure container within said second secure container.	The third secure container is a cabinet file signed by the application developer and including at least the licensed ActiveX control (first protected information. The licensing support code in the ActiveX control when its developer added licensing support to the ActiveX control is the third set of rules.
	24 A data managana amana amana ara in	Before an ActiveX control will create a
6	34. A data processing arrangement as in claim 33 further comprising means for	copy of itself, the calling application has to
7	applying said third set of rules to govern at least one aspect of use of said portion of	pass a license key to the ActiveX control. The license support code in the ActiveX
8	said first protected data.	control (third rule set) evaluates the authenticity of the calling application's
9		request.
10		
•	35. A data processing arrangement as in claim 34 further comprising means for	Windows Installer operating system service enforces the conditional syntax statements
11	applying said second set of rules to govern	of the application's signed .msi file. These statements govern the offer/installation of
12	at least one aspect of use of said portion of said first protected data.	the ActiveX control.
13		· · ·
14		
15		
16	•	•
17		
18		
19	•	
20		
21		·
		•
22	·	·
23		
24		
. 25		
26	·	
27	·	•
28		
	1	

	FOR U.S. PATENT NO. 5,915,019	
. 3	41	Infringing products include all Microsoft
4		tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the
5		Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
	A method comprising performing the	The signed .msi file created by the ActiveX control developer is the first secure.
7	following steps within a virtual distribution environment comprising one or more	container. The conditional syntax
8 9	electronic appliances and a first secure container, said first secure container comprising (a) a first control set, and	statement(s) of the ActiveX control developer's signed msi file is/are the first control set.
10	(b) a second secure container comprising a second control set and first protected	The first protected information is the ActiveX control.
11	information:	The first alternative for the second secure
12	•	container is the signed and licensed ActiveX control. The second control set is
13	ŗ	the license support code in the ActiveX control.
14		The second alternative for the second
15		secure container is a signed cabinet file containing the (signed or unsigned)
16	·	ActiveX control. The second control set is the license support code in the ActiveX control.
17 18	using at least one control from said first control set or said second control set to	The ActiveX control developer's conditional syntax statements (first control
19	govern at least one aspect of use of said first protected information while said first	set) in the ActiveX developer's signed .msi file govern the offer/installation of the
20	protected information is contained within said first secure container;	ActiveX control while it is in its signed .msi file.
21	-	Alternately, the license support code (second control set) in the ActiveX control
22	·	governs use of the licensed ActiveX control.
23	creating a third secure container comprising a third control set for governing	The third secure container is a signed .msi file. The application developer packages
24	at least one aspect of use of protected information contained within said third	its application in a signed .msi file (third secure container) and includes conditional
25	secure container;	syntax statements (third control set) in the signed .msi
26	incorporating a first portion of said first protected information in said third secure	Placing the ActiveX control into the application developer's signed .msi file
27	container, said first portion made up of some or all of said first protected	(third secure container).
28	information; and using at least one control to govern at least	The application developer's conditional
- 1	using at least one control to govern at least	ine application developer a conditional

	1	
1	one aspect of use of said first portion of said first protected information while said	syntax statement(s) in its signed .msi file govern the offer/installation ActiveX
2	first portion is contained within said third	control while it is in the signed .msi file (third secure container).
3	secure container.	(time secure contamer).
4	42. A method as in claim 41, in which said first secure container further includes a	The second protected information is a second. ActiveX control.
5	fourth secure container comprising a fourth	
6	control set and second protected information and further comprising the following step:	The first alternative for the fourth secure container is the signed and licensed second ActiveX control. The fourth control set is
7		the license support code in the ActiveX control.
- 8		The second alternative for the fourth secure
9		container is a signed cabinet file containing the (signed or unsigned) second ActiveX
10·		control. The fourth control set is the license support code in the ActiveX
11	using at least one control from said first	The ActiveX control developer's
12	control set or said fourth control set to govern at least one aspect of use of said	conditional syntax statements (first control set) in the ActiveX developer's signed .msi
13	second protected information while said second protected information is contained	file govern the offer/installation of the second ActiveX control while it is in its
14	within said first secure container.	signed .msi file.
15		Alternately, the license support code
16		(second control set) in the ActiveX control governs use of the licensed ActiveX control.
17		Control.
18	47. A method as in claim 41, in which said step of creating a third secure container	
19	includes: creating said third control set by	The application developer's conditional
20	incorporating at least one control not found in said first control set or said second	syntax statements are not found in either the first control set or the second control
21	control set.	set.
22	52. A method as in claim 41 in which said	
23	step of creating a third secure container occurs at a first site, and further comprising:	
24	comprising.  copying or transferring said third secure container from said first site to a second	The application developer at first site distributes its application to other sites.
25	site located remotely from said first site.	disdibutes to approation to outer sites.
26	53. A method as in claim 52 in which said	The application developer at the first site is
27	first site is associated with a content distributor.	the content distributor.
28	54. A method as in claim 53 in which said second site is associated with a user of	The application developer distributes the application to end-users.
	<b>.</b>	Exhibit B 113

1		
	content.	
2	55. A method as in claim 54 further	
. 3	comprising the following step: said user directly or indirectly initiating	For Internet downloads, the user initiates
. 4	communication with said first site.	the communication with the first site.
5	64. A method as in claim 54 in which said third control set includes one or more	The application developer's conditional syntax statements (third control set) govern
6	controls at least in part governing the use by said user of at least a portion of said	the installation of the ActiveX control (first protected information).
7	first portion of said first protected information.	
8		makida a sana a sahira isaha a sisaha
.9	76. A method as in claim 41 in which said creation of said third secure container further comprises using a template which	The third secure container is the application developer's signed .msi file and the third control set is the conditional syntax
10	specifies one or more of the controls contained in said third control set.	statements in that file.
· 11		Microsoft supplies several template .msi databases for use in authoring installation
12	·	packages. The UISample msi is the template recommended in the "An
13		Installation Example" on MSDN. This template msi files contains several default
14		conditional syntax statements. At least two of these conditional syntax statements
15		directly govern the installation by blocking progress until the EULA is accepted.
16	50	
17	78. A method as in claim 52 in which said creation of said third secure container further comprises using a template which	The third secure container is the application developer's signed .msi file and the third control set is the conditional syntax
18	specifies one or more of the controls contained in said third control set.	statements in that file.
19		Microsoft supplies several template .msi databases for use in authoring installation
20		packages. The UlSample msi is the template recommended in the "An
21		Installation Example" on MSDN. This template msi files contains several default
22		conditional syntax statements. At least two of these conditional syntax statements
23		directly govern the installation by blocking progress until the EULA is accepted.
24	·	· ·
25	·	
26		· · · · · · · · · · · · · · · · · · ·
27	,	•
28		
l	<b>]</b> •	

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

FOR U.S. 1A	1 EN1 NO. 3,313,017
81.	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft
	Installer technology.
A data processing arrangement comprising:  a first secure container comprising first	The first alternative for the first secure
protected information and a first rule set governing use of said first protected information;	container is the ActiveX control developer's signed .msi file containing a licensed ActiveX control (the first
,	protected information). The conditional syntax statements of the signed .msi file an
•	the first rule set.
	The second alternative for the first secure container is the signed cabinet file
	containing the ActiveX control. The license support code in the ActiveX control is the first rule set.
· ·	is the matruic sec
	The third alternative for the first secure container is the licensed and signed
· ·	ActiveX control governed by license support code in the ActiveX control.
a second secure container comprising a second rule set;	The second secure container is the signed .msi file which the application developer
	package its application. The second rule set is the conditional syntax statements of the application developer's signed .msi file
means for creating and storing a third	The third container is a signed cabinet file
secure container; and	containing at least the ActiveX control.
means for copying or transferring at least a portion of said first protected information	Putting the licensed ActiveX control (first protected information) in a signed cabinet
and a third rule set governing use of said portion of said first protected information to said second secure container, said means	file (third secure container). The licensing support code in the ActiveX control is third rule set.
for copying or transferring comprising: means for incorporating said third	Packaging the signed cabinet file in the
secure container within said second secure container.	signed .msi file.
92 A data	r
82. A data processing arrangement as in claim 81 further comprising:	The third rule set ensures the user is
means for applying at least one rule from said third rule set to at least in part govern at least one factor related to use of said	licensed.
portion of said first protected information.	]
83. A data processing arrangement as in claim 82 further comprising:	

Exhibit B

293482.02

1

means for applying at least one rule from said second rule set to at least in part govern at least one factor related to use of said portion of said first protected information.	The second rule set governs the offer/installation of first protected information.
	•

ا ت	· ·	<u>-</u>
. 4 .	85.	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the
5		Microsoft Installer SDK, and Operating System products that include the Microsoft
	4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Installer technology.
7	A method comprising the following steps: creating a first secure container comprising	The first protected information is the
8	a first rule set and first protected information;	ActiveX control.
9		The first alternative for the first secure container is the signed and licensed
10 11		ActiveX control. The first rule set is the license support code in the ActiveX control.
12		The second alternative for the first secure container is an (signed or unsigned)
13		ActiveX control with license support contained within a signed cabinet file. The
14	·	first rule set is the ActiveX license support code.
15	storing said first secure container in a first memory;	The first secure container is stored at the ActiveX control developer's location.
16	creating a second secure container	The second secure container is the application developer's signed .msi file.
17	comprising a second rule set;	The conditional syntax statements of the signed .msi file are the second rule set.
18	storing said second secure container in a	The second secure container is stored at the application developer's location.
19	second memory; copying or transferring at least a first	The ActiveX control developer packages
	portion of said first protected information	the control in a signed .msi file for
20	to said second secure container, said copying or transferring step comprising:	distribution to the application developer's site.
21	creating a third secure container	The third secure container is the ActiveX
22	comprising a third rule set;	control developer's signed .msi file containing a licensed ActiveX control. The
·		conditional syntax statements of the signed
23	gonving said first postion of said	.msi file are the third rule set. In preparation for using a msi authoring
24	copying said first portion of said first protected information;	tool, such as Microsoft's Orca, copying the ActiveX control to a package staging area.
25	transferring said copied first portion	Using msi authoring tool to import the
26	of said first protected information to said third secure container; and	control into the signed .msi file.
	copying or transferring said copied	The application developer installs the
27	first portion of said first protected information from said third secure	ActiveX control, which involves removing it from the ActiveX developer's signed
28	container to said second secure	msi file and installing it into its
	container.	environment. Subsequently, the
	1	47 · · ·

Exhibit B

293482.02

1

2

1		application developer places the ActiveX control into its signed .msi file when it is packaging its application.
3	87. A method as in claim 85 in which said	
4	copied first portion of said first protected information consists of the entirety of said	The entire ActiveX control is copied.
5	first protected information.	
6	89. A method as in claim 85 in which	I
7	said first memory is located at a first site,	The first memory is located at the ActiveX control developer's site.
8	said second memory is located at a second site remote from said first site, and	The second memory is located at the application developer's site.
9	said step of copying or transferring said first portion of said first protected information to said second secure container	The ActiveX control developer's signed msi file is transferred from its site to the site of the application developer.
0	further comprises copying or transferring	site of the application developer.
1	said third secure-container from said first site to said second site.	
2		•
3		
4	·	
5		
6		
7		
8		
9		• •
0		
1	·	
2		
3		
4		
5		
6	·	
27		
28		

3	FOR U.S. PATENT NO. 5,715,017	
4 5 6	85. (alternate infringing scenario)	Infringing products include all Microsoft tools that support the Microsoft ActiveX licensing model, Visual Studio .NET, the Microsoft Installer SDK, and Operating System products that include the Microsoft Installer technology.
7	A method comprising the following steps:	
8	creating a first secure container comprising a first rule set and first protected information;	The first protected information is the ActiveX control.
9	11101111111111,	The first alternative for the first secure container is the signed and licensed
10		ActiveX control. The first rule set is the license support code in the ActiveX:
11	·	control.
12		The second alternative for the first secure container is a (signed or unsigned) ActiveX
13 14		control with license support contained within a signed cabinet file. The first rule set would remain the ActiveX license
		support code.
15		The third alternative for the first secure container is a signed msi file in which the
16 17		ActiveX control developer packaged its ActiveX control. The first rule set is the conditional syntax statement(s) of the
18		signed msi file.
19	storing said first secure container in a first memory;	The first secure container is stored at the ActiveX control developer's location.
20	creating a second secure container comprising a second rule set;	The second secure container is the application developer's signed .msi file. The conditional syntax statements of the
21	storing said second secure container in a	signed .msi file are the second rule set.  The second secure container is stored at the
22	second memory;	application developer's location.
23	copying or transferring at least a first portion of said first protected information	The ActiveX control is placed in a cabinet file signed by the application developer and
24	to said second secure container, said copying or transferring step comprising:	the signed cabinet file is placed in a .msi file signed by the application developer.
	creating a third secure container	The third secure container is signed cabinet
25 26	comprising a third rule set;	file in which the application developer placed licensed ActiveX. The third rule set is the license support code in the ActiveX
27	copying said first portion of said	control. Copying ActiveX control.
	first protected information;	
28	transferring said copied first portion of said first protected information to	Transferring ActiveX control to signed cabinet file.
	1	日 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・

Exhibit B

1

1	said third secure container; and	
2	copying or transferring said copied first portion of said first protected	The application developer places the signed cabinet file into its signed .msi file when it
3	information from said third secure container to said second secure container.	is packaging its application.
4		
5	87. A method as in claim 85 in which said copied first portion of said first protected	The entire ActiveX control is copied.
6	information consists of the entirety of said first protected information.	
7	93. A method as in claim 85 in which	
8	said step of copying transferring said copied first portion of said first protected	The ActiveX control is placed in a cabinet file signed by the application developer and
9	information from said third secure container to said second secure container further comprises storing said third secure	the signed cabinet file is placed in a .msi file signed by the application developer.
10.	container in said second secure container.	
11		
12		
13 14		· ·
15	·	
16		
17		
18		
19		
20		
21	·	
22		
23		
24 25		
26		
27		
28		

3	FOR U.S. PATENT NO. 5,915,019	
. 4	1.	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET .CLR, and the Microsoft Installer
6		technology.
7	A method of operating on a first secure container arrangement having a first set of	The first protected content is a signed and licensed .NET component used by the
8	controls associated therewith, said first secure container arrangement at least in	NET assembly. The .NET assembly is distributed with a signed and governed .msi
9	part comprising a first protected content file, said method comprising the following	file. The second protected content is another signed and licensed .NET
10	steps performed within a virtual distribution environment including at least	component that is used by the .NET assembly.
11	one electronic appliance: using at least one control associated with said first secure container arrangement for	The first protected content is signed and licensed .NET component (first secure
12 13	governing, at least in part, at least one aspect of use of said first protected content	container) contained within the .NET assembly. The one control is a declarative
14	file while said first protected content file is contained in said first secure container	statement(s) within the assembly's header.
	arrangement;	The protected content is the same as the
15 16	creating a second secure container arrangement having a second set of controls associated therewith, said second	first protected content plus the additional implementation information included in the
17	set of controls governing, at least in part, at least one aspect of use of any protected	signed .msi file. The second secure container is the signed .msi file created for
18	content file contained within said second secure container arrangement;	the .NET assembly. The signed .msi file's conditional syntax statements are the
19	,	second set of controls that control the offer/installation of the .NET assembly.
20	transferring at least a portion of said first protected content file to said second secure	The entire .NET assembly is included in the signed .msi file.
21	container arrangement, said portion made up of at least some of said first protected content file; and	Packaging the .NET assembly in the signed .msi file involves the following process
22		steps. In preparation for using a msi authoring tool, such as Microsoft's Orca,
23		copying the .NET component to a package staging area. Using msi authoring tool to
24		import the .NET component into the signed .msi file.
25	using at least one rule to govern at least one aspect of use of said first protected content	The conditional syntax statement(s) of the signed msi file (second secure container)
26	file portion while said portion is contained within said second secure container	control(s) the offer/installation of the .NET assembly.
27	arrangement:	
28	said first secure container arrangement	The first alternative for the third secure
	comprises a third secure container	container is a licensed and signed NET
	· ·	· · · · ·

Exhibit B

293482.02

1

	<b>!</b>	•
2 . 3	arrangement comprising a third set of controls and said first protected content file, and	component governed by the set of declarative statements comprising the LicenseProviderAttribute (third set of controls).
5		The second alternative for the third secure container is a .NET component whose hash is included in the header of the .NET assembly. The set of declarative statements comprising the LicenseProviderAttribute is the third set of controls.
7 8 9	said first secure container arrangement further comprises a fourth secure container arrangement comprising a fourth set of controls and a second protected content file.	The first alternative for the fourth secure container is another licensed and signed .NET component governed by the set of declarative statements comprising the LicenseProviderAttribute (fourth set of controls).
11 12		The second alternative for the fourth secure container is the container created when the hash of the .NET component is included in the header information of the .NET assembly. The set of declarative
13 14		statements comprising the LicenseProviderAttribute is the fourth set of controls.
15		
16		
17		
		·
18		_
18 19		
19		
19 20 21 22		
19 20 21		
19 20 21 22 23 24		
19 20 21 22 23		
19 20 21 22 23 24		
19 20 21 22 23 24 25		

FOR U.S. PATENT NO. 5,915,019	
3 Infringing products include th	e .NET
Framework SDK, Microsoft \ NET, the Microsoft Installer	
products that include the Microsoft Instal	rosoft NET
6 A data processing arrangement comprising The first protected information	
at least one storing arrangement that at least temporarily stores a first secure	
container comprising first protected data and a first set of rules governing use of said container is the signed .msi file	le in which
first protected data,  the .NET component developed its .NET component. The first is the conditional syntax state signed .msi file.	t set of rules
The second alternative for the	first secure
container is a licensed and sig component governed by the s	ned .NET et of
declarative statements compri LicenseProviderAttribute of t	he:NET
component (first set of contro	
The third alternative for the fi is a signed cabinet file contain or unsigned). NET component	ning a (signed
support. The first set of control of declarative statements com LicenseProviderAttribute of t component.	prising the
and at least temporarily stores a second The second protected data is t	
secure container comprising second protected data different from said first protected data and a second set of rules assembly developer's assembly developer	
governing use of said second protected data; and data and a second set of roles  The first alternative for the se container is a signed .msi file	
22 NET assembly developer page multi-file assembly (second p	ckaged its
data). The second set of rules conditional syntax statements	s is the
.msi file that governs the offe of the .NET assembly.	
The second alternative for the	
secure container is a signed .? assembly. The second set of r	ules is the
declarative rules within the as	ssembly's
a data transfer arrangement, coupled to at least one storing arrangement, for assembly governed by declar	a signed .NET

Exhibit B

•		
1	transferring at least a portion of said first	its header (third set of rules). An
2	protected data and a third set of rules	alternative third rule set is the set of
2	governing use of said portion of said first	declarative statements comprising the
3	protected data to said second secure	LicenseProviderAttribute. The .NET
3	container,	assembly includes the .NET component.
A	·	The secure .NET assembly is included in a
. 4		signed .msi file (second secure container).
_		
5		An alternative third secure container is the
6		container created by hashing the .NET
O	•	component and including the hash in the
7		header information of a .NET assembly.
•		The .NET component is included in the
8		signed and governed .NET assembly
		(second secure container). The third set of
9		rules is the set of declarative statements
•		comprising the LicenseProviderAttribute.
10		
•		An alternative third secure container is a
11		signed cabinet file containing the .NET
		component and which is destined for a signed .msi file (second secure container).
12		The third set of rules is the set of
		declarative statements comprising the
13		LicenseProviderAttribute.
14	further comprising	
14	means for creating and storing, in said at	The first alternative for the third secure
15	least one storing arrangement, a third	container is a signed .NET assembly. In
1.5	secure container;	this case, the second secure container is the
16		signed .msi file.
	·	
17		The second alternative for the third
		container is the container created by
18		including a hash of the .NET component in
٠ .		the header information of a .NET assembly. In this case, the second secure container is
19		either the signed .msi file or the signed
		.NET assembly.
20		in assumory.
2,	•	The third alternative for the third container
21		is a cabinet file signed by the .NET
22		assembly developer containing the .NET
22		assembly and/or the .NET component. In
23	·	this case the signed .msi file is the second
رے		secure container.
24	said data transfer arrangement further	The first alternative for the third secure
	comprising means for transferring said	container is the signed .NET assembly,
25	portion of said first protected data and	which includes and/or uses the licensed
ŀ	said third set of rules to said third secure	NET component (first protected
26	container, and means for incorporating	information). The third set of rules is a
·	said third secure container within said	declarative rule within the .NET
27	second secure container.	assembly's header. The .NET assembly is
[		placed in a signed .msi file (second secure
28		container).
		. **

1 2 3 4 5		The second alternative for the third secure container is the container that results when the hash of the .NET component is added to the .NET assembly header information. The third set of rules is the set of declarative statements comprising the LicenseProviderAttribute added to the assembly.
6		The third alternative for the third secure
7		container is a cabinet file signed by the .NET assembly developer containing the
8	·	NET assembly and/or the .NET component. The third set of rules is a
9		declarative rule(s) within the .NET assembly's header and/or the set of
10		declarative statements comprising the LicenseProviderAttribute added to the
11		assembly
12	34. A data processing arrangement as in claim 33 further comprising means for applying said third set of rules to govern at	When the third rule set is the declarative statement(s) of the assembly header, the runtime CLR enforces the statements.
13	least one aspect of use of said portion of	When the third set of rules is the set of
14	said first protected data.	declarative statements comprising the LicenseProviderAttribute added to the
15		assembly, the license support code in the .NET component evaluates the authenticity of the calling assembly's request.
16		
17	35. A data processing arrangement as in claim 34 further comprising means for	When the second set of rules is the conditional syntax statements of the signed
18	applying said second set of rules to govern at least one aspect of use of said portion of	.msi file, the Windows Installer operating system service enforces the conditional
19	said first protected data.	syntax statements of .NET assembly's signed .msi file, which govern the
20		offer/installation of the .NET component.
21	·	When the second set of rules is the declarative statement(s) within the
22		assembly's header, the runtime CLR enforces the statements.
23	<del></del>	
24		•
-25		
26		·
27		
28	•	

-	FOR U.S. PAT	TENT NO. 5,915,019
3	41.	Infringing products include the .NET
4		Framework SDK, Microsoft Visual Studio NET, the Microsoft Installer SDK, and
5		products that include the Microsoft .NET CLR, and the Microsoft Installer
6	A method comprising performing the	technology. The signed .msi file created by the .NET
7	following steps within a virtual distribution environment comprising one or more	component developer is the first secure container. The first conditional syntax
8	electronic appliances and a first secure container, said first secure container comprising (a) a first control set, and	statement(s) of the .NET component developer's signed .msi file is/are the first control set.
10	(b) a second secure container comprising a second control set and first protected	The first protected information is the .NET component.
11	information:	The first alternative for the second secure
12		container is the signed and licensed .NET component. The second control set is the
13		set of declarative statements comprising the LicenseProviderAttribute.
14	·	The second alternative for the second secure container is a signed cabinet file.
15		The second control set remains the set of declarative statements comprising the
16		LicenseProviderAttribute.
17	using at least one control from said first control set or said second control set to	The .NET component developer's conditional syntax statements (first control
18	govern at least one aspect of use of said first protected information while said first	set) in its signed .msi file governs the offer/installation of the .NET component while it is in the signed .msi file.
19	protected information is contained within said first secure container;	
20		Alternately, the set of declarative statements comprising the
21		LicenseProviderAttribute (second control set) of the licensed .NET component
22	creating a third secure container	governs use of the .NET component.  The first alternative for the third secure
23	comprising a third control set for governing at least one aspect of use of protected	container is a signed .NET assembly, the protected information is the .NET
24	information contained within said third secure container;	component and the third control set is the declarative statement(s) within the .NET
25		assembly's header.
26		The second alternative for the third secure container is a signed .msi file in which the
27		NET assembly developer packages its NET assembly and the third control set is
28		the conditional syntax statement(s) in the signed msi file.
		1

Exhibit B

1	·	
1	incorporating a first portion of said first	In the first alternative, placing the .NET
2	protected information in said third secure container, said first portion made up of	component into the signed .NET assembly.
3	some or all of said first protected	In the second alternative, placing the .NET
	information; and	component into the. Net assembly developer's signed msi file.
4	using at least one control to govern at least	In the first alternative, the .NET assembly
5	one aspect of use of said first portion of said first protected information while said	developer's declarative statement(s) within the .NET assembly's header govern(s) the
6	first portion is contained within said third	use of the .NET component while it is in
	secure container.	the signed .NET assembly.
7		In the second alternative, the conditional
8		syntax statements of the .NET assembly developer's signed .msi file govern the
<b>.9</b> .		offer/installation of the .NET component
10		while it is in the signed msi file.
1	42. A method as in claim 41, in which said	The second protected information is a second .NET component.
11	first secure container further includes a fourth secure container comprising a fourth	•
12	control set and second protected information and further comprising the	The first alternative for the fourth secure container is the signed and licensed second
13	following step:	.NET component. The fourth control set is
		the set of declarative statements comprising the LicenseProviderAttribute of the second
14		.NET component.
15		The second alternative for the fourth secure
16	•	container is a second signed cabinet file.  The fourth control set is the set of
17		declarative statements comprising the
	using at least one control from said first	LicenseProviderAttribute. The .NET component developer's
18	control set or said fourth control set to	conditional syntax statements (first control
19	govern at least one aspect of use of said second protected information while said	set) in its signed .msi file governs the offer/installation of the second .NET
20	second protected information is contained	component while it is in the signed .msi
21	within said first secure container.	file.
	,	Alternately, the set of declarative
22		statements comprising the LicenseProviderAttribute (fourth control
23		set) of the licensed second .NET component governs use of the second .NET
24		component governs use of the second .INE1
	43. A mathed as in claim 41 in which said	
25	47. A method as in claim 41, in which said step of creating a third secure container	
26	includes:	The NET assembly developer's declaration
27	creating said third control set by a incorporating at least one control not found	The .NET assembly developer's declarative statements (first alternative for third control
	in said first control set or said second	set) and/or the developer's conditional
28	control set.	syntax statements (second alternative for the third control set) are not found in either

	·	
1		the first control set or the second control
2		set.
3	52. A method as in claim 41 in which said step of creating a third secure container occurs at a first site, and further	
5	comprising:  copying or transferring said third secure container from said first site to a second site located remotely from said first site.	The .NET assembly developer at first site distributes its assembly to other sites.
6	she located femotery from said first site.	
7 <sub>.</sub> 8	53. A method as in claim 52 in which said first site is associated with a content distributor.	The .NET assembly developer's business module is used to create and distribute its assembly.
9	54. A method as in claim 53 in which said second site is associated with a user of	The .NET assembly developer distributes the assembly to end-users.
10	content.	
11	55. A method as in claim 54 further comprising the following step:	
12	said user directly or indirectly initiating communication with said first site.	For Internet downloads, the user initiates the communication with the first site.
13 14	64. A method as in claim 54 in which said third control set includes one or more	When the third control set is the .NET assembly developer's declarative
15	controls at least in part governing the use by said user of at least a portion of said	statement(s) within the .NET assembly's header, it governs the user's use of the
16	first portion of said first protected information.	NET component (first protected information).
17		When the third control set is the .NET assembly developer's conditional syntax
18		statements of the .NET assembly developer's signed .msi file, it governs the
19		user's offer acceptance/installation of the .NET component (first protected information).
20		
21 22	76. A method as in claim 41 in which said creation of said third secure container further comprises using a template which	When the third secure container is the .NET assembly developer's signed .msi file and the third control set is the conditional
23	specifies one or more of the controls contained in said third control set.	syntax statements in that file.
24		Microsoft supplies several template .msi databases for use in authoring installation
25.		packages. The UlSample.msi is the template recommended in the "An
26		Installation Example" on MSDN. This template msi files contains several default
27		of these conditional syntax statements
28		directly govern the installation by blocking progress until the EULA is accepted.
		i i

1	
2	
. 3	
4.	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	-
23	
24	
25	
26	
~~	ŀ

78. A method as in claim 52 in which said creation of said third secure container further comprises using a template which specifies one or more of the controls contained in said third control set.

When the third secure container is the .NET assembly developer's signed .msi file and the third control set is the conditional syntax statements in that file.

Microsoft supplies several template .msi databases for use in authoring installation packages. The UISample.msi is the template recommended in the "An Installation Example" on MSDN. This template msi files contains several default conditional syntax statements. At least two of these conditional syntax statements directly govern the installation by blocking progress until the EULA is accepted.

27

	FUR U.S. FA	1EN1 NO. 5,315,013
3	81.	Infringing products include the .NET
4		Framework SDK, Microsoft Visual Studio
`		NET, the Microsoft Installer SDK, and products that include the Microsoft NET
5	<b>.</b>	CLR, and the Microsoft Installer
6		technology.
١	A data processing arrangement comprising:	m c NET
7	a first secure container comprising first protected information and a first rule set	The first protected information is the .NET component.
8	governing use of said first protected	
	information;	The first alternative for the first secure container is the signed .msi file in which
9		the .NET component developer packaged
10		its assembly. The first rule set is the
,,	·	conditional syntax statements written by the .NET component developer and placed
11	·	into the signed .msi file.
12		The second alternative for the first secure
13		container is the signed cabinet file
		containing the (signed or unsigned) .NET component. The set of declarative
14		statements comprising the
15		LicenseProviderAttribute when its
,,		developer added licensing support to the assembly is the first rule set.
16		
17		The third alternative for the first secure container is the licensed and signed .NET
18		component governed by the set of
		declarative statements comprising the LicenseProviderAttribute (first rule set)
19		added by the .NET component developer.
20	a second secure container comprising a	The first alternative for the second secure container is the signed .msi file in which
21	second rule set;	the .NET assembly developer packaged its
		NET assembly. The second rule set is the
22		conditional syntax statements written by the .NET assembly developer and placed
23		into the signed .msi file.
,		The second alternative for the second
24		secure container is the signed .NET
25		assembly. The second rule set is the declarative statements in the .NET
26		assembly's header.
	means for creating and storing a third	When the second secure container is the
27	secure container; and	signed msi file, the third secure container is the signed .NET assembly.
28		
		When the second secure container is the

Exhibit B

1

	<b>)</b>	
1		signed NET assembly, the third secure
2		container a .NET component secured by placing it in a signed cabinet file or by
. 3		including its hash in the header of the assembly.
. 4	means for copying or transferring at least a	When the second secure container is the
5	portion of said first protected information and a third rule set governing use of said portion of said first protected information	signed msi file and the third secure container is the signed .NET assembly, the third rule set is the set of declarative
6	to said second secure container, said means for copying or transferring comprising:	statements within the assembly's header.
7		When the second secure container is the signed NET assembly, the third rule set is
8		the set of declarative statements comprising the LicenseProviderAttribute (third rule
9		set) added to the .NET component by its developer.
10 -	means for incorporating said third secure container within said second	When the second secure container is the signed msi file and the third secure
11	secure container.	container is the signed .NET assembly, the assembly is placed in the signed .msi file.
12 13		When the second secure container is the signed .NET assembly and the third secure
13		container is a .NET component contained in a signed cabinet file or a .NET
15		component whose hash is included in the header of the assembly, the third secure container is incorporated within the .NET
16		assembly.
17	82. A data processing arrangement as in claim 81 further comprising:	
18	means for applying at least one rule from said third rule set to at least in part govern	When the third rule set is declarative statements within the assembly's header, it
19 20	at least one factor related to use of said portion of said first protected information.	governs the use of the .NET assembly which includes the first protected information.
21		When the third rule set is the set of
22	·	declarative statements comprising the LicenseProviderAttribute added by the .NET component by its developer, it
23		ensures the user is licensed.
24	83. A data processing arrangement as in claim 82 further comprising:	
25	means for applying at least one rule from said second rule set to at least in part	When the second rule set is the conditional syntax statements written by the .NET
26	govern at least one factor related to use of said portion of said first protected	assembly developer and placed into the signed .msi file, it governs the
27	information.	offer/installation of the .NET component.
28		When the second rule set is the declarative statements in the .NET assembly's header.

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

٠,		
3	85. A method comprising the following	Infringing products include the .NET
4	steps:	Framework SDK, Microsoft Visual Studio
		NET, the Microsoft Installer SDK, and
5		products that include the Microsoft .NET
		CLR, and the Microsoft Installer
6	ti	technology.  The first protected information is the .NET
_	creating a first secure container comprising	component.
7	a first rule set and first protected information;	component.
	iniomation,	The first secure container is a signed .NET
8		component (first protected information)
9		governed by the set of declarative
1		statements comprising the
10		LicenseProviderAttribute (first rule set).
11		The second alternative for the first secure
1		container is a cabinet file signed by the
12	•	NET component developer containing a
_		(signed or unsigned) .NET component with license support. The first rule set is the set
13	<u>'</u>	of declarative statements comprising the
		LicenseProviderAttribute.
14	storing said first secure container in a first	The first secure container is stored at the
15	memory:	.NET component developer's location.
•	creating a second secure container	The first alternative for the second secure
16	comprising a second rule set;	container is a signed NET assembly and
	•	the second rule set is declarative
17.		statement(s) within the assembly's header.
		The second alternative for the second
18		secure container is the signed .msi file in
		which the .NET assembly developer
19		packages its (signed or unsigned)
20		assembly. The second rule set is the
20		conditional syntax statement(s) written by
21		the .NET assembly developer and placed
		into the signed .msi file.
22	storing said second secure container in a	The second secure container is stored at the
_	second memory;	.NET assembly developer's location.
23	copying or transferring at least a first	The .NET component developer packages
ا بر	portion of said first protected information	its module in a signed .msi file for
24	to said second secure container, said	distribution to the .NET assembly developer's site.
25	copying or transferring step comprising: creating a third secure container	The third secure container is the signed
ر ۲	comprising a third rule set;	.msi file in which the .NET component
26	comprising a unitarate set,	developer packaged its .NET component.
20	·	The third control set is the conditional
27		syntax statements written by the .NET
		component developer and placed into the
28		signed msi file.
	copying said first portion of said	In preparation for using a msi authoring
n		::

Exhibit B

293482.02

	/	
l	first protected information;	tool, such as Microsoft's Orca, copying the .NET component to a package staging area.
2	transferring said copied first portion	Using the msi authoring tool to import the
3	of said first protected information to said third secure container; and	.NET component into the signed .msi file.
4	copying or transferring said copied first portion of said first protected	The .NET assembly developer installs the .NET component, which involves
5	information from said third secure	removing it from the .NET component
6	container to said second secure container.	developer's signed msi file and installing it into its environment. Subsequently, the
		.NET assembly developer places the .NET component into its .NET assembly and/or
7		signed .msi file when it is packaging its
8		NET assembly.
9	87. A method as in claim 85 in which said	The entire .NET component is copied.
	copied first portion of said first protected information consists of the entirety of said	' '
10	first protected information.	
11	00 1 1 1 1 1 1 1 1	<u> </u>
12	89. A method as in claim 85 in which said first memory is located at a first site,	The first memory is located at the .NET
		component developer's site.
13	said second memory is located at a second site remote from said first site, and	The second memory is located at the .NET assembly developer's site.
14	said step of copying or transferring said	The .NET component developer's signed .msi file is transferred from its site to the
15	first portion of said first protected information to said second secure container	site of the .NET assembly developer.
16	further comprises copying or transferring said third secure container from said first	
	site to said second site.	
17	94. A method as in claim 85 further	
18	comprising:	
19	creating a fourth rule set.	When the second secure container is not a signed .NET assembly, the fourth rule set is
•	•	declarative statements within the
20	•	assembly's header.
21		When the second secure container is not the signed .msi file in which the .NET
22		assembly developer packages its (signed or unsigned) assembly, the fourth rule set is
23		the conditional syntax statements written
24		by the .NET assembly developer and placed into the signed .msi file.
25		
26		
27		
28		
		я
		;; ;;

#### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

. 3		
	85 (alternate infringing scenario)	
4 .	A method comprising the following steps:	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and
6		products that include the Microsoft .NET CLR, and the Microsoft Installer technology.
7	creating a first secure container comprising a first rule set and first protected	The first protected information is the .NET component.
8	information;	The first alternative for the first secure
9		container is the signed and licensed .NET component. The first rule set is the set of
10		declarative statements comprising the LicenseProviderAttribute in the .NET component.
11		
12		The second alternative for the first secure container is a (signed or unsigned) .NET component with license support contained
13 14		within a cabinet file signed by the .NET component developer. The first rule set is
15		the set of declarative statements comprising the LicenseProviderAttribute in the .NET component.
16		The third alternative for the first secure
17		container is the signed .msi file in which the .NET component developer packaged
18		its assembly. The first rule set is the conditional syntax statements written by
19		the NET component developer and placed into the signed msi file,
20	storing said first secure container in a first	The first secure container is stored at the .NET component developer's location.
21	creating a second secure container	The first alternative for the second secure container is a signed NET assembly and
22	comprising a second rule set;	the second rule set is declarative
23		statement(s) within the assembly's header.
24.		The second alternative for the second secure container is the signed .msi file in
		which the .NET assembly developer packages its (signed or unsigned)
25	·	assembly. The second rule set is the
26		conditional syntax statement(s) written by the .NET assembly developer and placed
27.		into the signed .msi file.  The second secure container is stored at the
28	storing said second secure container in a second memory:	NET assembly developer's location.
<b>20</b> .	copying or transferring at least a first	The NET assembly developer places the
1		li li

Exhibit B

293482.02

1	···	
1	portion of said first protected information	.NET component into the third secure
2	to said second secure container, said copying or transferring step comprising:	container, which is either a signed cabinet file or a signed .NET assembly.
3	creating a third secure container comprising a third rule set;	When the second secure container is the signed .msi file, the third secure container
4		is the signed .NET assembly. The third rule set is the declarative statement(s) in
5		the .NET assembly's header.
6		When the second secure container is either a .NET assembly or the signed .msi file, the
7	·	third secure container is a signed cabinet file in which the .NET assembly developer
8	'	placed licensed .NET component. The third rule set is the set of declarative
9		statements comprising the LicenseProviderAttribute in the .NET
10		Component.
11	copying said first portion of said first protected information;	Copying the .NET component to either the .NET assembly or to the signed cabinet file.
12	transferring said copied first portion	Transferring the .NET component to either
12	of said first protected information to	the .NET assembly or the signed cabinet
13	said third secure container; and	file.
	copying or transferring said copied	When the second secure container is the
14	first portion of said first protected	signed .msi file and the third secure
1.5	information from said third secure container to said second secure	container is the signed .NET assembly, the .NET assembly is placed into the signed
15	container.	msi file.
1.6		
		When the second secure container is either the .NET assembly or the signed .msi file
17	·	and the third secure container is the signed
18	•	cabinet file, the signed cabinet file is placed
	· ·	into either the .NET assembly or the signed
19		msi file.
20	87. A method as in claim 85 in which said	The entire .NET component is copied.
20	copied first portion of said first protected	The chine is the component is copied.
21	information consists of the entirety of said	
	first protected information.	L
22	02 A	
23	93. A method as in claim 85 in which said step of copying transferring said	When the third secure container is the
23	copied first portion of said first protected	signed .NET assembly, it is placed in the
24	information from said third secure	signed .msi file.
	container to said second secure container	
25	further comprises storing said third secure	When the third secure container is a signed
26	container in said second secure container.	cabinet file, it can be placed in either the .NET assembly and/or the signed .msi file.
20		I assembly and the signed this me.
27	94. A method as in claim 85 further	
	comprising:	
28	creating a fourth rule set.	When the second rule set is declarative statement(s) within the assembly's header,

1 2		the fourth rule set is the conditional syntax statement(s) written by the .NET assembly
3		developer and placed into the signed .msi file.
4.		When the second rule set is the conditional
5		syntax statement(s) written by the .NET assembly developer and placed into the signed .msi file, the fourth rule set is
6	· · ·	declarative statement(s) within the assembly's header or the set of declarative
7		statements comprising the LicenseProviderAttribute in the .NET
8		component.
9	95. A method as in claim 94 further comprising:	
10	using said fourth rule set to govern at least one aspect of use of said copied first	If the fourth rule set is the .NET assembly developer's declarative statement(s) within
11	portion of said first protected information.	the .NET assembly's header, it governs the use of the .NET component.
12		If the fourth rule set is the conditional
13		syntax statements of the .NET assembly developer's signed .msi file, it governs the
14	·	offer/installation of the .NET component.
15		
16		
1,7.		
17. 18		
18		
18 19		
18 19 20		
18 19 20 21		
18 19 20 21 22		
18 19 20 21 22 23		
18 19 20 21 22 23 24		
18 19 20 21 22 23 24 25		
18 19 20 21 22 23 24 25 26		

Exhibit E

3	FOR U.S. PATEINT NO. 5,915,019		
4 5	85 (second alternate scenario for .NET)	Infringing products include the .NET Framework SDK, Microsoft Visual Studio .NET, the Microsoft Installer SDK, and products that include the Microsoft .NET CLR, and the Microsoft Installer technology.	
•	A method comprising the following steps:	John Marie Land	
7 8	creating a first secure container comprising a first rule set and first protected information;	The first protected information is a .NET component.	
		The first alternative for the first secure	
. 9	1.	container is the signed and licensed .NET	
. 10		component. The first rule set is the set of declarative statements comprising the LicenseProviderAttribute in the .NET	
11		component.	
12	1	The second alternative for the first secure	
13		container is a (signed or unsigned) .NET component with license support contained within a cabinet file signed by the .NET	
14		assembly developer. The first rule set is the set of declarative statements comprising	
15 16		the LicenseProviderAttribute in the .NET component.	
10		The third alternative for the first secure	
17		container is a .NET component whose hash	
18		is included in the assembly header of a .NET assembly. The first rule set is the set of declarative statements comprising the	
19		LicenseProviderAttribute in the .NET	
20		component.	
20	storing said first secure container in a first	The first secure container is stored at the	
21	memory;	.NET assembly developer's location.	
22	creating a second secure container comprising a second rule set;	The second secure container is the signed msi file in which the NET assembly	
23		developer packages its signed assembly.  The second rule set is the conditional	
24		syntax statement(s) written by the .NET assembly developer and placed into the	
		signed .msi file.	
25	storing said second secure container in a second memory;	The second secure container is stored at the .NET assembly developer's location.	
26	copying or transferring at least a first	The .NET assembly developer s location.	
[	portion of said first protected information	.NET component into the third secure	
27	to said second secure container, said	container, which is the signed .NET	
28	copying or transferring step comprising: creating a third secure container	assembly.	
20	comprising a third rule set;	The third secure container is a signed .NET assembly and the third rule set is	
	"	assembly and the time full set is	

Exhibit B

293482.02

1

•	· ·	
1		declarative statement(s) within the assembly's header.
2	copying said first portion of said first protected information;	Copying the .NET component to the .NET assembly.
. 3	transferring said copied first portion	Transferring the .NET component to the
4	of said first protected information to said third secure container; and	.NET assembly.
5	copying or transferring said copied	When the second secure container is the
6	first portion of said first protected information from said third secure	signed .msi file and the third secure container is the signed .NET assembly, the
7	container to said second secure container.	.NET assembly is placed into the signed .msi file.
8	87. A method as in claim 85 in which said	The entire .NET component is copied.
<b>.</b>	copied first portion of said first protected	The child it var component is copied.
. 9	information consists of the entirety of said first protected information.	e e e e e e e e e e e e e e e e e e e
10	90. A method as in claim 85 in which	
11	said first memory and said second memory are located at the same site.	First and second memory is at the .NET assembly developer's location.
12	are located at the same site.	assembly developer's location.
13	93. A method as in claim 85 in which	
14	said step of copying transferring said copied first portion of said first protected	When the third secure container is the signed .NET assembly, it is placed in the
15	information from said third secure container	signed .msi file.
16	further comprises storing said third secure container in said second secure container.	
17		
18	·	
		·
19		
20		
21		·
22		
23		
24.		
25		
26		·
27		
28		ii

#### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART

INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 5,915,019

3	<b>,</b>	•
٠	96. A method comprising performing the	A signed and licensed .NET component
4.	following steps within a virtual distribution	(first container) is part of a .NET assembly (second container), which is packaged in a
5	environment comprising one or more electronic appliances and a first secure	signed .msi file (third container).
3	container, said first secure container	uigited iiiid (aiii d o o o o o o o o o o o o o o o o o
6	comprising a first control set and first	
_	protected information:	T C
7	using at least one control from said first	The first secure container is a licensed and signed NET component governed by the
	control set to govern at least one aspect of use of said first protected information	set of declarative statements comprising the
. 8	while said first protected information is	LicenseProviderAttribute (one control).
9	contained within said first secure container;	
	creating a second secure container	The second secure container is a .NET
10	comprising a second control set for	assembly, the protected information is the
i	governing at least one aspect of use of	assembly and the second control set is declarative statement(s) within the
11	protected information contained within said second secure container;	assembly's header.
12	incorporating a first portion of said first	Included in the .NET assembly is the .NET
	protected information in said second secure	component.
13	container, said first portion made up of	
	some or all of said first protected information:	
14	using at least one control to govern at least	The declarative statement(s) govern the use
15	one aspect of use of said first portion of	of the .NET component and the custom
	said first protected information while said	LicenseProvider class (first control set)
16	first portion is contained within said second	controls the .NET component.
,,	secure container; and	The third secure container is the signed
17	incorporating said second secure container containing said first portion of said first	The third secure container is the signed .msi file in which the .NET assembly
18	protected information within a third secure	developer packages its assembly. The third
	container comprising a third control set.	control set is the conditional syntax
19		statements written by the assembly
		developer and placed into the signed .msi
20		file.
21		•

21

2

22 23

24

25

26

· 27

### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP.

2	INTERTRUST INFRINGEMENT CHART	
	[]	J.S. PATENT NO. 5,949,876
3	roke	7,0,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7
4.		
5	2.	Infringement is based on Microsoft's Visual Studio .NET and/or the .NET Framework licensing tools (in
6	·	the NET Framework SDK) and/or Microsoft Installer SDK
7	A system for supporting electronic commerce including:	
8	means for creating a first secure control set at a first location;	The first location is a .NET component developer's site.
9	set at a first location,	The first secure control set is the set of declarative statements comprising the License Provider Attribute of
10		a first .NET licensed component that provides for a
11	•	design-time license to use the control. This attribute also specifies the type of license validation that occurs.
,,		The component is encapsulated in a signed .NET assembly.
12 13	means for creating a second secure control set at a second location;	The second location is the .NET application developer's site where a .NET application comprising
1	Jonnes 30: 2: 2 3300.10 1032.1011,	one or more assemblies is created.
14		The second secure control set comprises the declarative statement(s) (including licensing
15		statements, and code access security statements) of a signed .NET assembly using or calling the first .NET
16		component. The control set can include a set of security permissions demanded by the .NET assembly
17		containing the licensed component, whereby the permissions are demanded of components that call the
18		application components. The control set can also be extended by controls expressed as conditional syntax
19		statements in a signed .msi file containing a click through end-user license (the end-user license
20		scenario).
21	means for securely communicating said first secure control set from said first	The first .NET control set is securely communicated from the first location developer to the .NET solution
22	location to said second location; and	provider by either being contained in a signed assembly, within a signed cabinet file or within a
23	means at said second location for	signed .msi file.  At the second location, the solution developer uses the
24	securely integrating said first and second control sets to produce at least a	.NET runtime that includes the LicenseManager.
25	third control set comprising plural	Whenever a class (control or component) is instantiated (here, an instance of the first .NET
26	elements together comprising an electronic value chain extended	licensed component), the license manager accesses the
27	agreement.	proper validation mechanism for the control or component. A value chain is created through the
28		creation of a run-time license for use of the first .NET component in the context of use of the .NET
20		application developed at the second location. The
- 1	· .	<del> </del>
- 1	•	

293482.02

Exhibit B

	. 1		license controls for the runtime license (derived from the design time license) are bound into the header of
	2		the .NET application assembly, along with the second
	3		control set.
	. 4		The creation of runtime license controls is securely handled by Visual Studio.NET or the LC tool.
	5		Runtime licenses are embedded into (and bound to) the executing assembly. The license control attribute
	6		included in the first .NET component is customized in the second location to express and require the runtime
	· 7	·	license. In a different scenario, the LC tool is used to create a ".licenses file" containing licenses for
	8		multiple components, including runtime licenses for components and classes created by the license
	9		provider. This .licenses file is embedded into the
	10		The third control set is an extended value chain
	11		agreement that comprises the runtime license controls for the first NET licensed class (that had been bound
	12		to the assembly), the declarative controls provided by the solution provider in the solution provider's
	13		assembly, and any runtime licenses for other components included by the solution provider in the
••	14		solution provider's assembly, and any end user license agreement provided by the application provider. The
	15	·	controls are typically integrated into the header of the .NET application assembly calling the first .NET
	16		licensed component.
	17		A further "end user licensing scenario" occurs when, at the second location, the application developer
	18	·	packages the application into a signed .msi file that includes conditional syntax statement controls that
		·	require that a user read and agree to an end user license agreement for the application and the
•	19		embedded first component. The third control set includes a plurality of elements that include the run-
	20		time licenses mentioned above, security permissions controls, EULA controls (a fourth control set), all
	21		securely bound into the signed .msi file.
	22		
	23	11. A system as in claim 2 in which said first location and said second location are	The Microsoft .NET Framework provides a Virtual Distribution Environment. Here the
	24	contained within a Virtual Distribution	nodes are the Common Language Runtime
	25	Environment.	instances that interpret the controls contained within .NET assemblies (among
•	26		other functions).
	27		
	28	29. A system as in claim 2 in which said first secure control set includes required	The licensing control in the first control set specifies the method required to validate
293482.02			Exhibit B
		,	•

terms.	the license.
32. A system as in claim 2 in which said second secure control set includes required terms.	The security permissions demanded (as described above) are required terms for execution of the application code element
60. A system as in claim 2 in which said means for securely integrating said first and	In the scenario where the application assembly is distributed using a signed .ms
second control sets includes a fourth	file, the secure integration of the first and second control sets is enhanced by the
condor set.	tamper protection afforded by the signed .msi file. In the end user license scenario,
	fourth control set consisting of conditions syntax statements is included in the .msi file.
	inc.
130. A system as in claim 2 further including means for executing said third control set within a protected processing	The third control set is executed under the auspices of the CLR
environment.	
122. A system as in claim 120 in which	The third control set is executed at an end
132. A system as in claim 130 in which said protected processing environment is located at a location other than said second	user site within the CLR.
location.	<u> </u>
161. A system as in claim 2 in which said	In the end user license scenario, the third
third control set includes controls containing human-language terms	control set includes a fourth control set the requires that the human user agree with
corresponding to at least certain of the machine-executable controls contained in	license terms displayed to the user. These human readable terms are referenced in the
said third control set.	conditional syntax statement controls contained in the signed .msi file.
162. A method as in claim 161 in which	The .msi file is a data descriptor data
said human-language terms are contained in one or more data descriptor data	structure.
structures.	
100	Im
170. A system as in claim 2 in which said means for creating a first secure control set includes a protected processing	The creation of the first licensed component, including its licensed control is carried out under the auspices of the
environment.	CLR.
171. A system as in claim 2 in which said	The application design time environment
means for creating a second secure control set includes a protected processing	and the creation of the .NET application i carried out under the auspices of the CLR
environment.	omittee out ander the auspites of the CEN
172. A system as in claim 2 in which said means at said second location for securely	The means for integrating the runtime license with the application controls is
integrating includes a protected processing environment.	carried out under the auspices of the CLR
329. A system as in claim 2 in which said	VS.NET runs under Windows.
	ii

	means for creating a first secure control set	
2	includes an operating system based on or compatible with Microsoft Windows.	
3	330. A system as in claim 2 in which said	VS.NET runs under Windows.
4	means for creating a second secure control	S. S. Talls didd! Willdows.
5	set includes an operating system based on or compatible with Microsoft Windows.	
6		VS.NET runs under Windows.
. 7		
8	sets includes an operating system based on or compatible with Microsoft Windows.	
9	346. A system as in claim 2 further	The third control set in the scenario
10	comprising means by which said third	described in the claim map for claim 2
11	one load module.	designed to be loaded into the CLR environment (a CLR host).
	242	
12	347. A system as in claim 2 farther comprising means by which said third	The third control set in the scenario described in the claim map for claim 2
13	control set governs the execution of at least one method.	governs a .NET executable. This
14		executable contains one or more methods.
15	349. A system as in claim 2 further comprising means by which said third	The third control set in the scenario described in the claim map for claim 2
15 16	349. A system as in claim 2 further comprising means by which said third control set governs the execution of at least one procedure.	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This
16	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23 24	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23 24 25	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more
16 17 18 19 20 21 22 23 24 25 26	comprising means by which said third control set governs the execution of at least	described in the claim map for claim 2 governs a .NET executable. This executable contains one or more

# INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,112,181

3		
4	CLAIM LANGUAGE	CLAIM OF INFRINGEMENT
5	48.	Infringing products include Microsoft SMS (Systems Management Server) 2.0 and subsequent versions.
6	A method for narrowcasting selected digital information to specified	
8	recipients, including:  a) at a receiving appliance, receiving selected digital information from a sending appliance remote from the	The receiving appliance is the client (e.g., end user computer in an Enterprise setting) receiving digital information (packages and/or
9	receiving appliance,	advertisement files) from the sending appliance, the centralized SMS database via a Client Access Point and/or Distribution Point set up on a server.
11 12	the receiving appliance having a secure node and being associated	The "node" is "secure" as a result of SMS security, as well as how it identifies and selects
13	with a specified recipient;	clients. The "specified recipient" is the result of the collection identifying a specific client that
14 15		meets the criteria for a package or advertisement.
		The desired in the configuration of the configurati
16 17	i) the digital information having been selected at least in part based on the digital information's membership in	The digital information is a software package or advertisement. The "first class membership was determined in part using rights management information" reads on creating
18 19	a first class, wherein the first class membership was determined at least in part using rights management information; and	software packages (or advertisements) based on attributes of the software.
20	ii) the specified recipient having been selected at least in part based on	The "specified recipient" is the client selected to receive a package or advertisement. That
21	membership in a second class, wherein the second class membership was	recipient is chosen based on a collection rule, or on the recipient's possession of a license.
22 23	determined at least in part on the basis of information derived from the specified recipient's creation, use of, or	
24	interaction with rights management information; and	
25	b) the specified recipient using the receiving appliance to access the	The receiving appliance is the client computer. The SMS agents on the client computer
26	received selected digital information in accordance with rules and controls, associated with the selected digital	receive, evaluate and take the appropriate action based on rules and controls governing the package and/or advertisement (i.e. the
27	information,	selected digital information).
28	should be a sife as i	Rules and controls are enforced by Agents on
	the rules and controls being enforced	Anies and controls are emoleced by Agents on

Exhibit B 145

1

		•
1	by the receiving appliance secure node.	the client (the secure node)
2		·
3	59. The method of claim 48 wherein	Event information includes SMS event
4	said received selected digital information is at least in part event information.	information, including Scheduling Classes.
5	63. The method of claim 48 wherein	All SMS packages must include a minimum of
6	said received selected digital information is at least in part executable software.	one program.
′	70. The method of claim 48 wherein	A control governs whether a MIF
8	said rules and controls at least in part govern usage audit record creation.	(management information file) is sent back to the SMS db after installation is done to report
9		on the success or failure of the installation.
10-	89. The method of claim 48 wherein said receiving appliance is a personal	The primary purpose of SMS is to manage
		software on personal computers throughout the
11	computer.	Enterprise.
11 12		
12		
12 13		
12 13 14		
12 13 14 15		
12 13 14 15		
12 13 14 15 16 17		
12 13 14 15 16 17		

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,112,181

. 3		
. 4	48.	Infringing products include Windows
. 5,		Media Player and Windows Media Rights Manager
6	A method for narrowcasting selected digital information to specified recipients,	This claim pertains to Windows Media Player with Individualized DRM Client and
7	including:	Windows Media Rights Manager used in the context of a narrowcast pay-per-view
8		(hear) media distribution service., simulcast and/or subscription services.
.9	(a) at a receiving appliance, receiving	Receiving appliance is a user's PC with
- 10	selected digital information from a sending appliance remote from the receiving	individualized DRM client (secure node). Specified recipient is a user using the
11	appliance, the receiving appliance having a secure node and being associated with a	specific individualized DRM client to access and render narrowcast pay-per-view
12	specified recipient	media, simulcast and/or subscription services for which the user acquires a
13	·	license.
14	(i) the digital information having been	The digital information is media that is
15	selected at least in part based on the digital information's membership in a first class,	narrowcast to licensed recipients. These narrowcast streams are licensed to users
. 16	wherein the first class membership was determined at least in part using rights	who have acquired licenses and whose PCs (appliances) support WMPs that have
17	management information; and	individualized DRM clients. This attribute is included in the signed WMA file header
18	·	and is used in the process of acquiring licenses for access to the media. Media that
19		are licensed to the recipient have their licenses bound to the recipient's
20	(ii) the specified recipient having been	Individualization module.  The recipient is selected for this content
21	selected at least in part based on membership in a second class, wherein the	based on the fact that the recipient is a member of the class of recipients who have
22	second class membership was determined at least in part on the basis of information	a license for the narrowcast media and whose devices support WMP and
23	derived from the specified recipient's creation, use of, or interaction with rights	individualized DRM clients. The recipient's machine must indicate support
24	management information; and	for individualization in challenges that are sent as part of requests for media in this
25	(b) the specified recipient using the	narrowcast class.  Recipient's machine uses WMP and the
26	receiving appliance to access the received selected digital information in accordance	individualized DRM client to access the narrowcast media in accordance with all
27	with rules and controls, associated with the selected digital information, the rules and	rules associated with the media and contained in the media license – in
28	controls being enforced by the receiving appliance secure node.	particular, requirements that individualization be supported.

Exhibit B

293482.02

#	
A CONTROL OF THE CONT	ACTUAL CONTRACTOR OF THE STATE
61. The method of claim 48 wherein said received selected digital information is at least in part entertainment information.	The digital information is Windows Media, which encodes audio/visual entertainment content.
62. The method of claim 61 wherein said entertainment information is at least in part music information.	Reads on narrowcast Windows Media Files that are music or audio/visual.
67. The method of claim 48 wherein said	The license contains a digital certificate.
rules and controls at least in part use digital certificate information.	The DRM client uses the certificate in the license to verify this signature and to verify that the header has not been tampered with.
72. The method of claim 48 wherein said rules and controls in part specifying at least one clearinghouse acceptable to rightsholders.	The signed header contains at least one URL that indicates to the Windows Media Rights Manager the license clearinghouse to be used in acquiring licenses.
75. The method of claim 72 wherein said at least one acceptable clearinghouse is a rights and permissions clearinghouse.	This clearinghouse is a license clearinghouse responsible for mapping rights and permissions onto requested content or narrowcasts and binding them to the requesting client environment or user of this environment.
89. The method of claim 48 wherein said receiving appliance is a personal computer.	Windows Media Player and the Individualized DRM client run on a personal computer.
	· ·.

#### INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART

2		VFRINGEMENT CHART TENT NO. 6,112,181
3	01	Infinite and the include Windows
4:	91	Infringing products include Windows Media Player and Windows Media Rights Manager
5	A method for securely narrowcasting selected digital information to specified recipients including:	This claim pertains to Windows Media Player with Individualized DRM Client and Windows Media Rights Manager used in
6 7	recipients including.	Windows Media Rights Manager used in the context of a narrowcast simulcast, payper-view (hear) media distribution service.
8		and/or subscription services. The content is delivered in a Protected Windows Media File.
<b>9</b> .	(a) receiving selected digital information in	Narrowcast content is received in a
10	a secure container at a receiving appliance remote from a sending appliance, the	Protected Windows Media File. Receiving appliance is user's PC with individualized
11	receiving appliance having a secure node, the receiving appliance being associated	DRM client (secure node).
12	with a receiving entity (i) the digital information having	The digital information is media that is
13	been selected at least in part based on the digital information's	narrowcast to licensed recipients (for example, a sold-out concert is narrowcast
14	membership in a first class,	on the Internet to "the class of" licensed (or ticketed) viewers).
15	(ii) the first class membership having been determined at least in	These narrowcast streams are licensed to users who have acquired licenses and
16	part using rights management information	whose PCs (appliances) support WMPs that have individualized DRM clients. This
17		attribute is included in the signed WMA file header and is used in the process of
18		acquiring licenses for access to the media.  Media that are licensed to the recipient
19		have their licenses bound to the recipient's individualization module.
20	(b) the receiving entity having been selected at least in part based on said	The recipient is selected for this content based on the fact that the recipient is a
21	receiving entity's membership in a second class,	member of the class of recipients who has a license for the narrowcast media.
22	(i) the second class membership having been determined at least in	The recipient class is determined by the license bound to the user's device that
23	part on the basis of information derived from the recipient entity's	supports WMP and individualized DRM clients. The recipient's machine must
24	creation, use of, or interaction with	indicate support for individualization in
25	rights management information	challenges that are sent as part of requests for media in this narrowcast class.
26	(c) receiving at the receiving appliance rules and controls in a secure container.	Receives a protected Windows Media File
27	(i) the rules and controls having been associated with the selected	Receives a license that is bound to the file as well as to the specific DRM client
	digital information; and	individualization information.
28	(d) using at the receiving appliance the selected digital information in accordance	Recipient's machine uses WMP and the individualized DRM client to access the

Exhibi B

293482.02

	with the rules and controls,	narrowcast media in accordance with all
2	.	rules associated with the media and
3		contained in the media license – in particular, requirements that
3		individualization be supported.
4	(i) the rules and controls being	The WMP and DRM client enforce the
5	enforced by the receiving appliance secure node.	rules embedded in the Protected Windows Media File License.
,		
. 6	104. The method of claim 91 wherein said received selected digital information	The digital information is Windows Media,
7	includes entertainment information.	which encodes audio/visual entertainment content.
8	109. The method of claim 91 wherein said rules and controls at least in part use digital	The DPM client was the DPM client was the DPM client was the contains a digital certificate.
. 9	certificate information.	The DRM client uses the certificate in the license to verify this signature and to verify
10	<del> </del>	that the header has not been tampered with.
. 10	114. The method of claim 91 wherein said	The signed header contains at least one
11	rules and controls specify at least one	URL that indicates to the Windows Media
12	clearinghouse acceptable to rightsholders.	Rights Manager the license clearinghouse to be used in acquiring licenses.
13	117. The method of claim 114 wherein said at least one acceptable clearinghouse is a	This clearinghouse is a license
14	rights and permissions clearinghouse.	clearinghouse responsible for mapping rights and permissions onto requested
		content or narrowcasts and binding them to
15	<b>i</b> .	content of harroweasts and billottig them to
15		the requesting client environment or user of
15 16		the requesting client environment or user of this environment.
	131. The method of claim 91 wherein said receiving appliance is a personal computer	the requesting client environment or user of this environment.  Windows Media Player and the
16 17	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.
16	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22 23	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22 23	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22 23 24 25	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22 23 24 25 26	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22 23 24 25	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a
16 17 18 19 20 21 22 23 24 25 26	131. The method of claim 91 wherein said receiving appliance is a personal computer.	the requesting client environment or user of this environment.  Windows Media Player and the individualized DRM client run on a

## INTERTRUST TECHNOLOGIES CORP. v. MICROSOFT CORP. INTERTRUST INFRINGEMENT CHART FOR U.S. PATENT NO. 6,389,402

· · · · · · · · · · · · · · · · · · ·	
HER OF CLAIMITANGUAGE THE SELECTION OF T	CEAIM OF INFRINGEMENTS
1.	Products infringing: Microsoft Visual Studio .NET, .NET License Compiler, .NET Framework SDK, and .NET Common Language Runtime
A method including	A method for producing a third .NET component (application) that incorporates firs and second .NET component whose
	distribution is license controlled.
creating a first secure container including a first governed item and having associated a first control;	The first secure container is a first signed NET component that includes a license control. The governed item is the .NET
	component.
	The first control is the set of declarative statements comprising the LicenseProviderAttribute of a first .NET
:	licensed component that provides for a designime license to use the control. This attribute also specifies the type of license validation the
·	occurs.
creating a second secure container including a second governed item and having associated a second control;	The second secure container is the second signed .NET component that includes a licen control. The governed item is the .NET component.
	The second control is the set of declarative statements comprising the License Provider Attribute of a second .NET
	licensed component that provides for a desig time license to use the control. This attribute also specifies the type of license validation the occurs.
transferring the first secure container from a first location to a second location;	The creator distributes a signed and licensed .NET component.
	An application developer at a second location downloads a first .NET component for inclusion into an application.
transferring the second secure container from a third location to the second location;	A creator distributes a signed and licensed .NET component from a different location.
	Application developer downloads a second .NET component for inclusion into an application.

Exhibit B

293482.02

		•
ì		
2	at the second location, obtaining access to at least a portion of the first governed item, the access being governed at least in part by the first control;	At the second location, the application developer uses the .NET runtime that includes the LicenseManager to access a first governed item.
4		Whenever a class (control or component) is
5 6		instantiated (here, an instance of the first .NET licensed component), the license manager accesses the proper validation mechanism for the control or component.
7		The first control comprises the declarative
8 9	·	statement(s) (including licensing statements, and code access security statements) of the first .NET component.
y		
10	at the second location, obtaining access to at least a portion of the second governed item, the access being governed at least in part by the	At the second location, the application developer uses the .NET runtime that includes the LicenseManager to access a second
11	second control;	governed item.
12	·	Whenever a class (control or component) is instantiated (here, an instance of the second
13		.NET licensed component), the license manager accesses the proper validation
14	·	mechanism for the control or component.  The second control comprises the declarative
15 16		statement(s) (including licensing statements, and code access security statements) of the second .NET component.
	at the second location, creating a third secure	At the second location, the application
17 18	container including at least a portion of the first governed item and at least a portion of the second governed item and having associated at least one control, the creation being governed	developer uses the .NET runtime that includes the LicenseManager to access a first governed item and second governed item to construct an application, the third secure container.
19	at least in part by the first control and the	
20	second control.	Creation governance is accomplished by invoking the .NET runtime to access the first governed item and the second governed item.
21		Whenever a class (control or component) is
22		instantiated the license manager accesses the proper validation mechanism for the control or
23		component.
24		The portions of the first governed item and second governed item that are being included
25		in the third secure container will typically include the governed items themselves, ie. the
26		.NET components.
27		The associated control in this case is the LicenseProviderAttribute, created and inserted
28		into the application.

新の動物 おおな · 多人

#### EXHIBIT C

CONFIDENTIAL—SUBJECT TO PROTECTIVE ORDER OF NOVEMBER 19, 2001: Exhibit C contains documents or things that are the subject of a Protective Order of this Court and cannot be opened or its contents made available to anyone other than this C urt or counsel of record for the parties.